Free Flight Store Simulation
Using Beggar

THESIS

Judson T. Babcock, Second Lieutenant, USAF

AFIT/GAE/ENY/06-M02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/06-M02

Free Flight Store Simulation Using Beggar

THESIS

Presented to the Faculty
Department of Aeronautics and Astronautics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Judson T. Babcock, B.S.A.E.
Second Lieutenant, USAF

March 2006

AFIT/GAE/ENY/06-M02

Free Flight Store Simulation Using Beggar

Judson T. Babcock, B.S.A.E.
Second Lieutenant, USAF

Approved:

/signed/ 23 Mar 2006

Lt Col R.C. Maple (Chairman) date

/signed/ 23 Mar 2006

Maj R.J. McMullan (Member) date

/signed/ 23 Mar 2006

Maj P. Blue (Member) date

AFIT/GAE/ENY/06-M02

Abstract

The complete numerical solution of the airflow around a store in extended free flight is of particular importance to the United States Air Force. Beggar is the primary CFD program used by the USAF to obtain solutions for store separations. However, Beggar's ability to simulate a store in free flight is limited because the store must fall through a static background mesh, eventually reaching a point where the solution will fail. The length of any free flight simulation is consequently limited by the height of the background mesh. Code modifications are made to Beggar to remove this requirement by pinning the store in the background mesh at its center of gravity. Rotations are accomplished within the background mesh, but translations are reflected as changes in the grid speeds of the background mesh. This allows the numerical simulation to continue indefinitely. Beggar's ability to model moving components (e.g. control surfaces) in multi-body problems is fully preserved. The modified code is applied to the MK-84 AIR model, which demonstrates that the solution of a pinned store using the modified code adequately matches the solution of a translating store using the unmodified code. In addition, extended free flight simulations are conducted in which the dynamic behavior and long term trajectory of the store are observed. The longest simulation lasts for 135 seconds of solution time. Testing of a generic store body with multiple moving fins results in good agreement between the unmodified and modified solution methods. The modified code reduces overall computational cost by 17% for simulations of similar length because of the smaller background mesh. The combination of indefinite runtime and control surface modeling will make Beggar a powerful tool for studying the non-linear dynamic behavior of stores in free flight.

Acknowledgements

Thanks must first go to God, who has answered many prayers in my life, with the present accomplishment being no exception. I would also like to express my thanks to Lt Col Raymond Maple for his invaluable assistance and to Bruce Jolly and Magdi Rizk for freely giving of their time and expertise. To my family and friends, thank you for your prayers, support, and understanding during this onerous time.

Judson T. Babcock

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xiii

List of Symbols xiv

List of Abbreviations xv

I. Introduction 1
- 1.1 Research Goals 2
- 1.2 Prior Research 3
  - 1.2.1 Wind Tunnel Techniques 3
  - 1.2.2 Test & Evaluation Techniques 4
  - 1.2.3 Computational Techniques 5
  - 1.2.4 Beggar Development 6
  - 1.2.5 Rigid Body Motion 7
  - 1.2.6 Prior Validation 8
- 1.3 Research Approach 10
  - 1.3.1 Beggar Modifications 10
  - 1.3.2 Testing Approach 10
- 1.4 Document Organization 11

II. Computational Theory 12
- 2.1 Governing Equations 12
  - 2.1.1 Non-dimensionalization 14
  - 2.1.2 Flow Discretization 15
  - 2.1.3 Boundary Conditions 17
- 2.2 Overset Grids in Beggar 19
  - 2.2.1 Grid Hierarchy 20
  - 2.2.2 Grid Communication 21
- 2.3 Six Degree-of-Freedom Model 23
  - 2.3.1 Beggar Coordinate Systems 24
  - 2.3.2 Transformations in Beggar 26
  - 2.3.3 Dynamic Equations of Motion 31

Page

2.3.4 Equation of Motion Solution Method 35
2.4 Current Approach 35

III. Methodology 37
3.1 Beggar Modifications 37
3.1.1 Mesh Motion 37
3.1.2 Flow Visualization 38
3.1.3 Translation Eliminations 39
3.2 Supersonic Compression Ramp 41
3.3 MK-84 AIR 42
3.3.1 Grid Generation 43
3.3.2 Grid Dimensions 45
3.3.3 MK-84 AIR Testing 46
3.3.4 MK-84 AIR Numerical Validation 47
3.4 Generic Store with Moving Components 50
3.4.1 Generic Store Grids 50
3.4.2 Generic Store Testing 52
3.5 Beggar Inputs 53

IV. Results & Discussion 55
4.1 Supersonic Compression Ramp 55
4.1.1 Ramp Convergence History 55
4.1.2 Ramp Results 55
4.2 MK-84 AIR Results 58
4.2.1 Convergence History 58
4.2.2 Comparison Test Results 58
4.2.3 Free Flight Simulation Results 66
4.3 Generic Store Results 76

V. Conclusions 87
5.1 Future Research 88

Appendix A. Extended Results 89
A.1 Compression Ramp Case Results 89
A.2 MK-84 AIR Convergence History 90
A.3 MK-84 AIR Comparison Test Case 2 91
A.4 Generic Store Body Convergence History 94

Page

Appendix B. Beggar Input Files 96

B.1 MK-84 AIR Input File 96

B.2 MK-84 AIR Boundary Conditions 99

B.3 MK-84 AIR Time Step Ramping Schedule 99

B.4 Generic Store Body Specifications 100

Bibliography 105

List of Figures

Figure Page

1. Translating Method of Free Flight Store Simulation 2

2. Pinned Method of Free Flight Store Simulation 3

3. Block-to-Block, Patched, and Overlapping Grid Communications [1] 19

4. Tall Background Mesh surrounding MK-84 AIR body 21

5. Standard CFD Coordinate System 24

6. Euler Rotation Angles 27

7. Rotation about the x axis 28

8. Rotation/Translation Process in Beggar 29

9. Supersonic Compression Ramp 41

10. MK-84 General Purpose Bomb 43

11. MK-84 AIR Computational Model 43

12. MK-84 AIR Superblock 44

13. Small Background Mesh Surrounding MK-84 AIR Body 45

14. G&C Coordinate System 49

15. Generic Store Body Computational Model 50

16. Grid Cutplane around Generic Store Body with Fins in “X” configuration 51

17. Positive Rotation Direction of Upper Fins 52

18. Prescribed Fin Deflection vs. Time Profile for Upper Fins . . . 53

19. Convergence History of the Compression Ramp 56

20. Ramp Case 1: Mach 2.0 Flow, No Mesh Motion 56

21. Comparison of Mach number across the ramp with varying amounts of mesh motion 57

22. Force History on MK-84 AIR body at Mach 0.6 at 20,000 ft . . 60

23. Moment History on MK-84 AIR body at Mach 0.6 at 20,000 ft 60

Figure Page

24. MK-84 AIR Velocity History at Mach 0.6 at 20,000 ft 61

25. MK-84 AIR Angular Velocity History at Mach 0.6 at 20,000 ft 61

26. MK-84 AIR Trajectory at Mach 0.6 at 20,000 ft 62

27. MK-84 AIR Orientation at Mach 0.6 at 20,000 ft 62

28. Contours of Mach showing the MK-84 AIR in its pre-release position at Mach 0.6 at Sea Level 67

29. Mach History of Extended MK-84 AIR Simulations 68

30. Contours of Mach showing the MK-84 AIR just after entering the supersonic flight regime 69

31. Extended Force History on MK-84 AIR body at Mach 0.6 at Sea Level 69

32. Extended Moment History on MK-84 AIR body at Mach 0.6 at Sea Level 70

33. Extended Angular Velocity History of MK-84 AIR at Mach 0.6 at Sea Level 71

34. Extended Trajectory of MK-84 AIR at Mach 0.6 at Sea Level . 71

35. Extended Orientation History of MK-84 AIR at Mach 0.6 at Sea Level 72

36. Contours of Mach around the MK-84 AIR after 135 seconds of free flight 72

37. Extended Force History on MK-84 AIR body at Mach 0.9 at 10,000 ft 73

38. Extended Moment History on MK-84 AIR body at Mach 0.9 at 10,000 ft 74

39. Extended Velocity History of MK-84 AIR at Mach 0.9 at 10,000 ft 74

40. Extended Trajectory of MK-84 AIR at Mach 0.9 at 10,000 ft . 75

41. Extended Orientation History of MK-84 AIR at Mach 0.9 at 10,000 ft 76

42. Contours of Static Pressure over the Generic Store in Initial Pitch at Mach 0.6 at 20,000 ft 77

Figure Page

43. Force History on Generic Store Body at Mach 0.6 at 20,000 ft . 79

44. Moment History on Generic Store Body at Mach 0.6 at 20,000 ft 79

45. Generic Store Body Velocity History at Mach 0.6 at 20,000 ft . 80

46. Generic Store Body Angular Velocity History at Mach 0.6 at 20,000 ft 80

47. Generic Store Body Trajectory at Mach 0.6 at 20,000 ft 81

48. Generic Store Body Orientation History at Mach 0.6 at 20,000 ft 81

49. Extended Force History on Generic Store Body at Mach 0.6 at 20,000 ft 83

50. Extended Moment History on Generic Store Body at Mach 0.6 at 20,000 ft 84

51. Extended Velocity History of Generic Store Body at Mach 0.6 at 20,000 ft 84

52. Extended Angular Velocity History of Generic Store Body at Mach 0.6 at 20,000 ft 85

53. Extended Trajectory of Generic Store Body at Mach 0.6 at 20,000 ft 85

54. Extended Orientation History of Generic Store Body at Mach 0.6 at 20,000 ft 86

A.1. Ramp Case 2: Mach 1.5 flow and Mach 0.5 mesh motion 89

A.2. Ramp Case 3: Mach 0.5 flow and Mach 1.5 mesh motion 89

A.3. Convergence of Static Forces on MK-84 AIR 90

A.4. Convergence of Static Moments on MK-84 AIR 90

A.5. Force History on MK-84 AIR body at Mach 0.9 at 10,000 ft . . 91

A.6. Moment History on MK-84 AIR body at Mach 0.9 at 10,000 ft 91

A.7. MK-84 AIR Velocity History at Mach 0.9 at 10,000 ft 92

A.8. MK-84 AIR Angular Velocity History at Mach 0.9 at 10,000 ft 92

A.9. MK-84 AIR Trajectory at Mach 0.9 at 10,000 ft 93

A.10. MK-84 AIR Orientation at Mach 0.9 at 10,000 ft 93

Figure Page

A.11. Convergence of Static Forces on Generic Store Body 94

A.12. Convergence of Static Moments on Generic Store Body 95

List of Tables

Table Page

1. Phases of Beggar Code Modifications 10

2. Coordinate System Abbreviations 30

3. Compression Ramp Test Cases 42

4. MK-84 General Purpose Bomb Specifications 44

5. MK-84 AIR Grid Dimensions 45

6. MK-84 AIR Test Cases 47

7. Generic Store Grid Dimensions 51

8. Standard Deviations of Forces and Moments in the Final 200 Iterations of the Static Solution 58

9. MK-84 AIR Test Case 1: Forces and Moments after 2.36 seconds 64

10. MK-84 AIR Test Case 1: Velocities after 2.36 seconds 64

11. MK-84 AIR Test Case 1: Trajectory and Orientation after 2.36 seconds 64

12. MK-84 AIR Test Case 2: Forces and Moments after 2.27 seconds 65

13. MK-84 AIR Test Case 2: Velocities after 2.27 seconds 66

14. MK-84 AIR Test Case 2: Trajectory and Orientation after 2.27 seconds 66

15. Wall Clock Times for MK-84 AIR Test Cases 66

16. Standard Deviations of Forces and Moments in the Final 100 iterations of Generic Store Static Solution 76

17. Generic Store Body: Forces and Moments after 1.4 seconds . . 78

18. Generic Store Body: Velocities after 1.4 seconds 78

19. Generic Store Body: Trajectory and Orientation after 1.4 seconds 82

List of Symbols

Symbol		Page
$\mathcal{V}$	volume	13
$\vec{Q}$	vector of conserved variables	13
$\vec{F}_c$	convective flux vector	13
a_t	contravariant face speed	13
x_t, y_t, z_t	mesh speeds in the principle directions	13
U	contravariant velocity	13
E_t	total energy	14
$\hat{u}$	internal engergy	14
$\vec{F}_v$	viscous flux vector	14
τ	viscous stress tensor	14
α, β, γ	Euler angles	26
F	fin coordinate system	30
FCO	fin coplanar coordinate system	30
$FCOI$	fin coplanar initial coordinate system	30
G	global coordinate system	30
S	store coordinate system	30
ϕ_{F2FCOI}	fin deflection angle	30
CFL	Courant-Friedricks-Lewy Number	41
ϕ	roll angle	48
θ	pitch angle	48
ψ	yaw angle	48
u, v, w	velocity components in the principle directions	49
$\dot{\psi}, \dot{\theta}, \dot{\phi}$	angular velocities around the principle axes	49
M_2	Mach number behind shock wave	57
M	Mach number	63

List of Abbreviations

Abbreviation		Page
USAF	United States Air Force	1
CFD	Computational Fluid Dynamics	1
AFSEO	Air Force Seek Eagle Office	1
AFB	Air Force Base	1
(6+)DOF	six degree-of-freedom model	1
CG	center of gravity	3
CTS	Captive Trajectory System	4
ARDS	Advanced Range Data System	5
GPS	Global Positioning System	5
TSPI	Time-Space-Position-Information	5
T&E	Test & Evaluation	5
AFFTC	Air Force Flight Test Center	5
EnRAP	Enhanced Range Applications Program	5
IDAPS	Image Data Automated Processing System	5
CAT	Computational Aeromechanics Team	6
3D	three-dimensional	7
JDAM	Joint Direct Attack Munition	8
CBU	Cluster Bomb Unit	9
SMC	store moving component	9
SMB	store main body	10
RANS	Reynolds-Averaged Navier-Stokes	12
CM	center of mass	32
2D	two-dimensional	42
GP	general purpose	42
PVM	Parallel Virtual Machine	47
G&C	Guidance & Control	48

Free Flight Store Simulation Using Beggar

I. Introduction

The United States Air Force (USAF) has a large requirement for numerous, quick reaction solutions to the airflow around a wide variety of stores and aircraft. Knowing the aerodynamic characteristics of these stores prior to any flight testing can increase efficiency and reduce the danger of unpredicted store behavior during testing. Unfortunately, with the advent of increasingly complex and accurate weapons systems, it is getting more difficult to find a "truth source" test instrumentation system to properly analyze a store via traditional flight testing [32].

Accurate numerical solutions can provide this efficiency and safety while reducing costs when bringing a weapon through the development phase. The Beggar Computational Fluid Dynamics (CFD) code, developed and maintained by the Air Force Seek Eagle Office (AFSEO) at Eglin Air Force Base (AFB), Florida, was developed for this purpose. It combines a flow solver, six degree-of-freedom ((6+)DOF) integrator, and overset grid assembler to obtain the solution around the complicated geometries required for store separation simulations. However, its use to model the dynamic behavior of stores in free flight has been limited because of Beggar's requirement for an inertially fixed background mesh.

Currently, to simulate a store in free flight, Beggar assembles the store grid and an inertially fixed background mesh with its Chimera grid assembly system. The store is placed in its initial position near the top of the tall background mesh. The coupled flow solver and (6+)DOF integrator contained in Beggar allows the store to fall through the background mesh in response to the aerodynamic and gravitational forces acting on it. A simple illustration of this is shown in Figure 1. Eventually the store will reach a position where grid assembly fails as it moves past the bottom of the

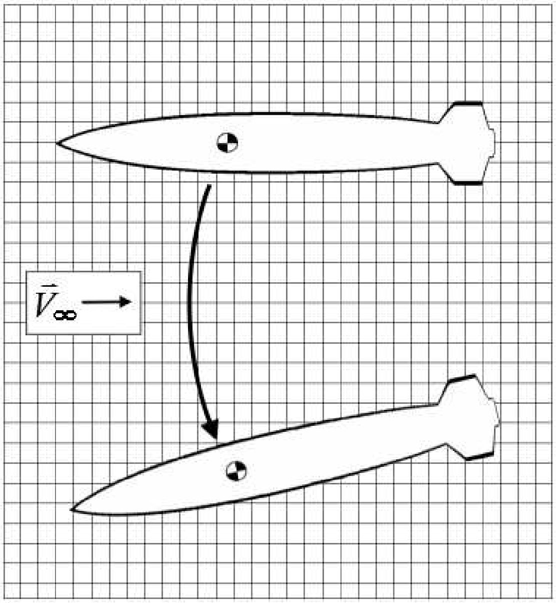

Figure 1: Translating Method of Free Flight Store Simulation

background mesh. Hence, the length of the free flight simulation in this case is limited by the height of the background mesh, impeding the use of Beggar for extended high-fidelity dynamic free flight simulations. Such a simulation would require a 'taller' background mesh, with sufficient density throughout the mesh for successful grid assembly. Because a longer free flight simulation will result in more stream-wise displacement from the initial x-location, the required size of the background mesh is increased yet again.

As the background mesh grows in response to these requirements for extended simulations, every iteration of the flow solver becomes more computationally expensive. This can result in very large computational problems with long run times and cumbersome memory requirements. These limitations prevent Beggar from simulating a store in extended free flight for such purposes as trajectory predictions or dynamic stability analysis.

1.1 Research Goals

The goal of this research is to modify the Beggar code to remove the requirement for an inertially-fixed background mesh, greatly improving the flexibility of the

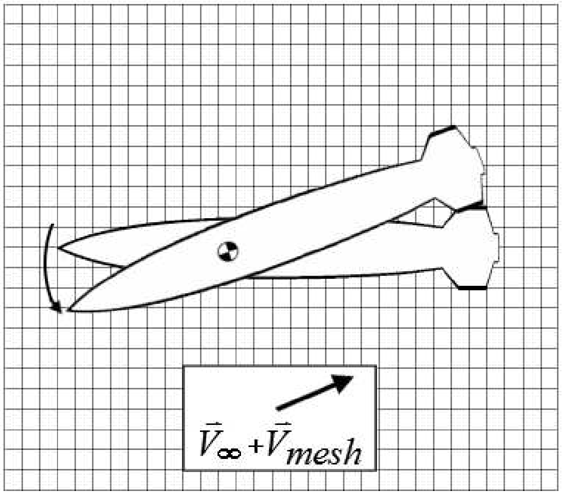

Figure 2: Pinned Method of Free Flight Store Simulation

Beggar code. Under the non-inertial simulation approach, the center of gravity (CG) of a single store maintains a fixed position relative to a smaller background mesh (Figure 2). As the simulation progresses, rotations about the CG will be modeled by rotating the store grids within the background mesh, but translations will be reflected as changes in the grid velocities of the background mesh. Since the store is no longer translating relative to background mesh, the simulation can proceed for an indefinite period of time without grid assembly failure. Because the background mesh is smaller, the overall computational cost is greatly reduced. In addition, the implementation of rotational store motion through grid motion simplifies the computation of background mesh velocities and preserves Beggars ability to model control surface functionality.

The combination of indefinite runtime and control surface modeling will make Beggar a powerful tool for studying the non-linear dynamic behavior of stores in free flight. The application of this research to the problem of predicting the dynamic stability characteristics of stores will be of great benefit to the U.S. Air Force.

1.2 Prior Research

1.2.1 Wind Tunnel Techniques. Before the advent of Computational Fluid Dynamics, pre-flight store certification was primarily performed through wind tunnel testing. Two main wind tunnel techniques exist to investigate store separation behavior. The Captive Trajectory System (CTS) uses a complicated setup of a sting-mounted store and aircraft model integrated with a computer controlled six degree-of-freedom model. The dynamic equations of motion and measured aerodynamic loads are used from one time step to calculate the motion of the store, which is moved relative to the aircraft by a computer-controlled mechanism [12]. The Captive Trajectory System can also be used to obtain the aerodynamics of the store alone, without the aircraft model and the associated interference effects. At the end of the separation event, data from the test can then be used in combination with a 6DOF model to determine the projected impact point of the weapon [38]. Although this is typically an accurate approach when validated with flight test data, sting interferences effects on the store carriage loads have been known to compromise the accuracy of the data in some situations [9].

The free drop method is the second wind tunnel method available for store release simulations. A free drop test is performed by placing the aircraft with attached store in free stream conditions in a wind tunnel and simply dropping the store model while recording the separation event [12]. The many disadvantages of this method outweigh any benefit derived from removing the sting and model support system from the flow field. The store model must be carefully constructed to prevent damage to the wind tunnel, but then is destroyed when dropped. These models are also expensive to manufacture because of the inertial characteristics of the store that need to be simulated in the model. Obtaining undistorted inertial properties within the model while preserving the correct center of gravity and center of mass locations is also exceptionally difficult. Ejection forces on the store are also extremely difficult to simulate by the free drop method.

1.2.2 Test & Evaluation Techniques. The trajectory and aerodynamic characteristics of stores can also be obtained through traditional flight testing, with the accuracy of the test instrumentation limiting the accuracy of the results. Some of these techniques can also be used to study store separation events, although a full scale test is obviously required. Current flight testing systems such as the Advanced Range Data System (ARDS) or radar, LASER, and optical tracking systems are quickly becoming less desirable with the advent of highly accurate weapons systems. Typical test instrumentation is expected to be one order of magnitude (10 times) more accurate than the weapon being tested [32]. For example, if a weapon is accurate to within 50 feet of a target, the test instrumentation used should be accurate to within 5 feet. This continually places new, more stringent demands on test instrumentation as more accurate weapons system are being fielded by the United States military. The Global Positioning System (GPS) led to the development of ARDS, which provided Time-Space-Position-Information (TSPI) with an unprecedented accuracy of 2-4 meters when it was fielded in the early 1990s. Several Test & Evaluation (T&E) ranges, such as the White Sands Missile Range and the Air Force Flight Test Center (AFFTC) range at Edwards AFB, are still using this system today.

The latest test instrumentation system, the Enhanced Range Applications Program (EnRAP), will be fielded in 2007. It provides sufficient accuracy and upgrade capability to continually meet the needs of the Army, Navy, and Air Force in the future [32]. This system will provide real-time position accuracy of 0.3 meters as well as a “GPS-denied” TSPI accuracy of 8-16 meters.

The current optical tracking system used at Eglin AFB, Florida uses high-speed cameras mounted on the test aircraft to determine the store trajectory after release. These cameras capture the store separation at 200 frames per second, which is then analyzed by technicians using the Image Data Automated Processing System (IDAPS) to obtain position and orientation information from the store throughout the separation event [17]. This method has been used as the primary tool to study store separation events.

1.2.3 Computational Techniques. As computational methods mature, more computational techniques are being integrated into the traditional test methodology. Considerable progress has been made in the accuracy of computational models (especially turbulence models) in recent years, which has lent credibility to computational methods and increased the demand for computational solutions. Assisting this progress has been the recent exponential growth in computing capability, especially processing power and memory capacity, allowing for increased problem sizes and accuracies. The current efforts pressing towards an even greater expansion of computing capability only beget a larger demand for computational solutions in the future.

1.2.4 Beggar Development. The Beggar code originated in 1994 at the U.S. Air Force Wright Laboratory at Eglin AFB, FL. It has been used exclusively by the Air Force Seek Eagle Office at Eglin AFB since then and continues to be under development by the Computational Aeromechanics Team (CAT) located there. Beggar was developed specifically to numerically resolve the complicated flow that exists around multiple geometries in relative motion, such as an aircraft/store combination. It does this by using blocked and overset grid techniques, with an automated grid assembly process, flexible flow solution, and (6+)DOF motion model.

The Beggar program has been used extensively by the Air Force to compute carriage loads and store separation trajectories. These capabilities have been continuously tested and validated while the code has been in production use. An overview of some of the validation efforts is presented in Section 1.2.6.

The development of the Beggar code is an ongoing process, guided mainly by the efforts of the Computational Aeromechanics Team at Eglin AFB. The goal of this development process is to upgrade the capability of the code in order to simulate the increasingly complex weapons systems being introduced by the U.S. Air Force [26]. Because of the continually changing state of Air Force weapons development, the Beggar code is perpetually being upgraded and extended. Another goal is to continually increase the computational efficiency of the code. Since the flow solver is still the most computationally intensive part of the code, new solution methods and turbulence models are always being evaluated.

1.2.5 Rigid Body Motion. As there is greater advancement in the range of applications for CFD, there have been many more CFD algorithms designed for moving meshes. The concept of moving meshes may be found in problems involving mesh deformation or rigid body motion, and especially if the problem involves unsteady flows. Mesh deformation problems include oscillating airfoils, adapting meshes, or even hydrodynamic problems with free surfaces. Such applications often involve stationary boundaries where the mesh only deforms inside the boundaries. Rigid body motion applications include cases where the mesh and boundaries move rigidly through the flow as a single body. Such cases include the presently studied problem of store separation or bodies in relative motion.

Rigid body motion has already been validated in these applications. Biedron, Vatsa, and Atkins showed that an unsteady flow past a pitching airfoil and pitching blended-wing body using mesh motion resulted in good agreement between computed and measured values of pressure coefficients and certain dynamic stability derivatives [5]. Their work also validated a more complex piston-driven synthetic jet with mesh motion, in which they compared the jet velocities with experimental data. Good agreement between experimental and computational data was seen in both the oscillating airfoil and three-dimensional (3D) wing tests in this research. Hughson developed a 3D unstructured method for dynamic motion which successfully predicts the unsteady solution about a pitching rectangular wing [16].

Beggar uses the concept of rigid body motion to apply varying grid speeds to bodies in relative motion. The validation of the mesh motion in Beggar is inherent to any discussion of the validity of a Beggar solution involving such motion. With that in mind, rigid body motion has not been applied or validated on a background mesh in the past. However, the same equations in the flow solver are used on the background mesh as are used on any other grid, and other grids have commonly been validated with components of mesh motion present.

1.2.6 Prior Validation. The research done by Coleman, Jolly, Chesser, and Brock analyzed Beggar's capability for a simple store separation event from an F-15E aircraft with the MK-84 general purpose bomb [10]. The store was released at a Mach number of 0.90 with the aircraft at an angle of attack of 1.1 degrees. The store trajectory was computed using an inviscid flow field and compared against wind tunnel data. The computed position of the center of gravity of the store showed excellent agreement with the wind tunnel data. The orientation of the store throughout the separation event also agreed with the wind tunnel data, with the exception of the store roll angle. The differences in roll angle were attributed to the way the ejector forces are modeled in Beggar versus in the wind tunnel.

Prewitt, Belk, and Maple investigated the accuracy of Beggar in varying test cases [25]. Their research addressed the accuracy of the Beggar numerical solution as compared against wind tunnel data for a store in the carriage position. Good agreement was found between Beggar and the wind tunnel data, with most discrepancies being attributed to the lack of viscous effects in the numerical solution. They also compared store trajectories from different test cases against known wind tunnel data, and found that Beggar's predictions had good agreement with the wind tunnel data. Their final test case consisted of ejecting three equivalent stores loaded on a triple ejector rack. The results from this case were similar to results by Thoms and Jordan [31]. In these differing cases, Beggar proved to be a robust and reliable method of predicting the behavior of stores.

A fully time accurate solution to the separation of a Joint Direct Attack Munition (JDAM) GBU-31 from an F-18C aircraft was presented by Noack and Jolly [22]. Two test cases were run, both with the aircraft in a dive at approximately 45 degrees nose down at transonic speeds (Mach 0.962 and Mach 1.055). Inviscid solutions were obtained for both Mach numbers, and a viscous solution was obtained for the Mach 0.962 case. Overall, the numerical solutions were found to agree well with flight observations. Some differences were found between the inviscid and viscous solutions, demonstrating that while having higher a computation requirement, the viscous solution does achieve greater accuracy. This test case validated the use of Beggar in the more complex, highly non-linear transonic store separation environment.

Rizk and Lee [27] demonstrated the accuracy of the (6+)DOF by using the Beggar code in the absence of a flow field. The solutions were checked by simply verifying that they satisfy the dynamic equations of motion. An unconstrained store body was used in combination with a single store moving component. These components were represented by a single rod of length 20.0 and a connected rod of length 8.0, respectively. The center of mass of each rod was located at its midpoint, and the combined system was aligned with the XY plane, resulting in motion within that plane only. The system was at an initial state of rest with an applied moment in the Z-direction for the first 2 seconds of motion. A constant force was also applied to the centers of mass of the components for this initial 2 second period. At the end of this period, both forces and moments were removed and the system was allowed to rotate freely.

The initial motion consisted of both components rotating and translating in the XY plane. After the initial period, the angular speed of each component became periodic, with the period being a function of the geometry and inertial properties of the system. The resulting motion of the system computed by the (6+)DOF was found to align exactly with the motion predicted by the equations governing the dynamic motion.

In the same research, Rizk and Lee also demonstrated the usefulness and accuracy of the (6+)DOF solver in the full scale simulation of a store separation from an F-16 aircraft. The store used was a CBU-89 Gator Mine, a 1,000 pound cluster munition containing anti-tank and anti-personnel mines. The CBU-89 was equipped with retractable fins, which are modeled as store moving components (SMC) in Beggar. These fins remain retracted prior to release, then deploy after the store is released. Dudley and Westmoreland had previously tested and validated the capability of the (6+)DOF to model these retractable fins as SMCs under prescribed motion [11]. The fins must be deployed by means of prescribed motion because of the absence of detailed

Table 1: Phases of Beggar Code Modifications.

Phase	**Objective**
Phase 1a	Application of mesh motion on the background mesh.
Phase 1b	Correction of flow visualization
Phase 2	Implementation of pinned SMB
Phase 3	Implementation of pinned SMB with SMCs

information (such as spring constants and inertial properties) about the mechanisms that deploy the fins. This setup by Rizk and Lee also tested the capability of the Chimera grid assembly method because of the overset grids used to represent the fins. The results of the test, a time history of x, y, and z displacement and θ, ϕ, and ψ angles of the munition, showed good agreement to known flight test data.

1.3 Research Approach

1.3.1 Beggar Modifications. The complete modification to the Beggar code will be broken down into multiple phases, shown in Table 1. First, modifications will be made to the Beggar code to enable the application of mesh motion to the background mesh. Next, the flow visualization will be corrected to include this component of mesh motion. The major modification will be removing the inertially fixed background mesh by pinning the store at its center of gravity in the background mesh. This will first be done for the simplest case: a single store body. Then the code will be adapted for the multi-body problem consisting of the store main body (SMB) with multiple fins modeled as moving components.

1.3.2 Testing Approach. Since mesh motion and the visualization of that motion are interdependent, the first testing will occur after Phase 1. This test is designed to be a simple confirmation that the mesh motion and flow visualization corrections are properly applied by using a supersonic compression ramp. The solution of any combination of mesh and flow motion over the ramp should result in a similar shock angle and flow field. These two measures of merit are easily verified against the exact solution obtained from the oblique shock relations.

The next testing occurs after Phase 2 through the simulation of MK-84 AIR single body store. A “tall” background mesh with extended vertical height is used to find the reference solution of the store translating in response to the gravitational and aerodynamic forces acting on it. A smaller background mesh is used to find the solution of the pinned store at the same initial conditions, using the modified code to remove the inertially fixed background mesh. A simple comparison is conducted between the two solution methods, and the results should match closely. Extended free flight simulations will also be accomplished with the MK-84 AIR to demonstrate the capacity of these modification.

Finally, testing of the multi-body problem will be completed. This testing will use a generic store body with multiple fins that rotate in response to a prescribed motion input by the user. Another simple comparison solution between the translating case and pinned case will be run.

1.4 Document Organization

The governing equations are presented in Chapter II, along with an overview of Beggar’s implementation of those equations. A detailed description of the overset grid capability and the (6+)DOF model used in Beggar is also given. Emphasis is placed on the internal methods Beggar uses for coordinate systems and transformations, since an understanding of these methods is crucial to this research. Chapter III presents an overview of the methodology used in this research, including the code modifications and specific test cases. Results are given in Chapter IV along with a discussion of the pertinent findings. Conclusions and recommendations for future research are presented in Chapter V. The appendices are used for the presentation of extended results and detailed methodology that are not discussed directly in Chapter III or IV but are relevant to the research.

II. Computational Theory

The Beggar code was developed specifically to address the flow interactions around multiple geometries in relative motion, such as a store separation events [19]. The flow in such a scenario can be extremely difficult to solve for a number of reasons. Complicated geometries such as the aircraft itself, stores, pylons, and weapons bays make grid generation difficult. These geometries introduce acoustic and aerothermodynamic flow phenomena which can be difficult to simulate numerically [29]. In addition, the store separation problem adds moving components to the flow solution which must be accounted for.

Beggar simplifies grid generation around these geometries through the use of blocked, patched, and overset grid techniques, which decompose these geometries into subdomains. The generation of the required structured grids is much simpler around these subdomains. Beggar combines this grid assembly with a versatile flow solver which solves the Navier-Stokes equations using multiple numerical schemes and iterative techniques [7]. A 6+ degree-of-freedom ((6+)DOF) model is coupled with this flow solver and allows for the simulation of moving bodies in response to forces and moments calculated by the flow solver [27]. After release, the trajectory and orientation of the store is computed and saved.

The combination of these techniques makes Beggar the foremost tool for such an analysis by the USAF. The effectiveness and accuracy of this approach has been proven over and over again through various tests conducted by the AFSEO [7], some of which have been shown in Section 1.2.6.

2.1 Governing Equations

Beggar is capable of obtaining a numerical solution to the Reynolds-Averaged Navier-Stokes (RANS) equations, the thin-layer Navier-Stokes equations, or the Euler equations [35]. The Euler equations may be obtained by simply removing the viscous terms from the Navier-Stokes equations, which are reviewed here. The RANS equations are not covered.

The Navier-Stokes equations refer to the system of equations comprised of the conservation laws of mass, momentum, and energy [6]. These conservation laws are derived for a Newtonian fluid in a continuum and apply to the flow of fluid through a finite control volume. The Navier-Stokes equations are the collection of these laws, show here in the integral form with no body forces:

$$\int_{\mathcal{V}} \frac{\partial \vec{Q}}{\partial t} d\mathcal{V} + \oint_{A} \left(\vec{F}_c - \vec{F}_v\right) dA = 0 \tag{1}$$

where $\mathcal{V}$ represents the volume of the cell. The vector of conservative variables $\vec{Q}$ in three dimensions is given by:

$$\vec{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix} \tag{2}$$

The inviscid, convective flux vector $(\vec{F}_c)$ at the surface of a control volume is defined as:

$$\vec{F}_c \cdot \hat{n} = \begin{bmatrix} \rho U \\ \rho u U + p n_x \\ \rho v U + p n_y \\ \rho w U + p n_z \\ (E_t + p)\vec{V} + p a_t \end{bmatrix} \tag{3}$$

where the motion of the mesh appears through the contravariant face speed (a_t), which is defined as:

$$a_t = x_t n_x + y_t n_y + z_t n_z \tag{4}$$

The x_t, y_t, z_t variables represent the mesh speeds in the x, y and z directions. The contravariant velocity (U) is written relative to the motion of the mesh and defined as:

$$U = \vec{V} \cdot \vec{n} = (u - x_t)\, n_x + (u - y_t)\, n_y + (u - z_t)\, n_z \tag{5}$$

Total energy (E_t) is defined as:

$$E_t = \rho \left(\hat{u} + \frac{1}{2} \left| \vec{V} \right|^2 \right) \tag{6}$$

where $\hat{u}$ is the internal energy.

The vector of viscous fluxes $(\vec{F}_v)$ contains the viscous stress in the three principle directions, as well as work and heat conduction terms:

$$\vec{F}_v = \begin{bmatrix} 0 \\ \tau_{xx} n_x + \tau_{xy} n_y + \tau_{xz} n_z \\ \tau_{yx} n_x + \tau_{yy} n_y + \tau_{yz} n_z \\ \tau_{zx} n_x + \tau_{zy} n_y + \tau_{zz} n_z \\ \Theta_x n_x + \Theta_y n_y + \Theta_z n_z \end{bmatrix} \tag{7}$$

The work done by the viscous stresses and heat conduction Θ_i is:

$$\Theta_i = u_j \tau_{ij} + k \frac{\partial T}{\partial x_i} \tag{8}$$

and the viscous stress tensor (τ) is defined from the deformation law for a Newtonian fluid:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \nabla \vec{V} \tag{9}$$

When combined with Stokes hypothesis $(\lambda + \frac{2}{3}\mu = 0)$, this reduces to:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \nabla \vec{V} \right) \tag{10}$$

2.1.1 Non-dimensionalization. As is customary, Beggar solves the non-dimensional form of the governing equations. This allows the characteristic parameters to be varied independently of the solution [30] and results in the flow variables being normalized according to Equation 11 [36], where the asterisks denote the non-dimensional variables. These equations are substituted into the governing Navier-Stokes equation to complete the non-dimensionalization process.

$$\begin{gathered} \rho^* = \rho/\rho_\infty \qquad E_t^* = E_t/\rho_\infty a_\infty^2 \qquad p^* = p/\rho_\infty a_\infty^2 \qquad t^* = ta_\infty/L_{ref} \\ u^* = u/a_\infty \qquad v^* = v/a_\infty \qquad w^* = w/a_\infty \end{gathered} \tag{11}$$

2.1.2 Flow Discretization. The Navier-Stokes equations accurately define the flow at all points in time. However, they are non-linear, non-unique, and complex [36]. Even today, relatively few exact solutions have been found, and those solutions deal with highly simplified problems. Therefore, a numerical solution to these equations must be obtained, meaning the governing equations must be discretized across the solution domain. An implicit time integration approach is advantageous because of the large time steps that can be used without damaging the stability of the solution [6]. This results in increased computational efficiency over explicit schemes. The Navier-Stokes equations shown in Section 2.1 are shown again here with an implicit discretization [18]:

$$\frac{Q^{n+1} - Q^n}{\Delta t}\mathcal{V} + \sum \left(\vec{F}_c^{\,n+1} - \vec{F}_v^{\,n+1} \right) = 0 \tag{12}$$

where the time is discretized with a first order backward Euler discretization [29] and the superscript n represents the current time step. Equation 12 states that the change in the conserved variables with time in the cell volume plus the sum of the fluxes through the cell boundaries must equal zero. These fluxes are linearized in time [18], resulting in:

$$F_c^{n+1} \approx F_c^n + \frac{\partial F_c}{\partial Q}\left(Q^{n+1} - Q^n\right) \tag{13}$$

$$F_v^{n+1} \approx F_v^n + \frac{\partial F_v}{\partial Q}\left(Q^{n+1} - Q^n\right) \tag{14}$$

Using these linearized fluxes in Equation 12 results in:

$$\frac{Q^{n+1} - Q^n}{\Delta t}\mathcal{V} + \frac{\partial R}{\partial Q}\left(Q^{n+1} - Q^n\right) = -\mathcal{R}^n \tag{15}$$

where the flux Jacobian $\frac{\partial R}{\partial Q}$ is defined as:

$$\frac{\partial R}{\partial Q} = \sum \left(\frac{\partial F_c^n}{\partial Q} - \frac{\partial F_v^n}{\partial Q}\right) \tag{16}$$

The left hand side of Equation 15 represents the implicit side, and the right hand side is the residual (explicit) side, defined as:

$$\mathcal{R}^n = \sum \left(\vec{F}_c^n - \vec{F}_v^n\right) \tag{17}$$

Beggar uses Newton sub-iterations at each time step to accurately compute unsteady flows. The user may specify the number of Newton iterations or a convergence criteria. For a generic system of equations given by $G(x) = 0$, Newton's method can be written as [37]:

$$G^{'}(x^m)(x^{m+1} - x^m) = -G(x^m) \tag{18}$$

where $G^{'}(x)$ is given by:

$$G'(x) = \begin{bmatrix} a_{11}(x) & a_{12}(x) & \cdots & a_{1n}(x) \\ a_{21}(x) & a_{22}(x) & \cdots & a_{2n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(x) & a_{n2}(x) & \cdots & a_{nn}(x) \end{bmatrix} \tag{19}$$

and

$$a_{ij}(x) = \frac{\partial G_i(x)}{\partial x_j} \tag{20}$$

Equation 15 may be rewritten in the form of Equation 18:

$$\left(\frac{\partial R}{\partial Q}\right)^{n,m}\left(Q^{n+1,m+1} - Q^{n+1,m}\right) = -\left[\frac{Q^{n+1} - Q^n}{\Delta t}\mathcal{V} + \mathcal{R}^n\right] \tag{21}$$

To solve this equation,the $Q^{n+1,m+1}$ term must be isolated. This is difficult because of the size of the Jacobian (a block-pentadiagonal matrix with dimensions on the order of millions). Beggar uses a symmetric Gauss-Seidel relaxation scheme to solve this. The Gauss-Seidel approach solves the generic equation $[A]\,x = b$ for x by dividing $[A]$ into

$$[A] = ([D]\,[-L])\,[-U] \tag{22}$$

where $[D]$, $[-L]$, $[-U]$ are the diagonal part of $[A]$ and the entries below and above the diagonal, respectively [20]. The following equation is then solved to obtain x:

$$x_i^k = \left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^k - \sum_{j=i+1}^{j_{\max}} A_{ij} x_j^{k-1} \right) / A_{ii} \tag{23}$$

These Gauss-Seidel iterations are the “inner” iterations as used in Beggar. For both steady-state and unsteady problems, the user may specify the number of inner iterations to use, or the iterations may be continued until one of the convergence criteria are met.

The flux Jacobians of Equation 16 are derived using first-order Steger-Warming fluxes, while the residual is derived using either second-order Roe flux difference splitting or second-order Steger-Warming flux vector splitting schemes [26]. The Steger-Warming approach splits the convective fluxes into their positive and negative parts according to the eigenvalues, while the Roe method is an approximate Riemann solver based on the decomposition of the difference in fluxes over a cell face [6].

2.1.3 Boundary Conditions. Boundary conditions are specified on one or more regions in Beggar. A region is a subset of a computational grid defined by the user, which may include a point, line, surface, or volume of a grid. There are a number of boundary condition types available including tangent (inviscid) wall, no slip (viscous) wall, farfield, overlap, as well as porous, mass flow, and heat source boundaries [1].

Explicit boundary conditions are the default in Beggar, although fully implicit boundary conditions can also be used. The implicit boundary conditions are updated during the Gauss-Seidel solution discussed above. An underrelaxation factor is applied to improve stability [25].

A tangent, or inviscid, boundary condition is the default surface boundary condition used in this research. This inviscid boundary defines a surface to be impermeable with the normal component of the velocity vector at the surface being zero. The effect is a completely tangential velocity vector at the surface. The values of the flow at the surface are found through the following equations, where the reference values are taken from the first cell in the computational domain:

$$\begin{aligned} p_{body} &= p_{ref} + \rho_{ref} a_{ref} U_{ref} \\ \rho_{body} &= \rho_{ref} + \left(p_{body} - p_{ref}\right) / a_{ref}^2 \\ u_{body} &= u_{ref} - U_{ref} n_x \\ v_{body} &= v_{ref} - U_{ref} n_y \\ w_{body} &= w_{ref} - U_{ref} n_z \end{aligned} \tag{24}$$

where the mesh motion is included in the U_{ref} term, which equals:

$$U_{ref} = \left(u_{ref} - x_t\right) n_x + \left(v_{ref} - y_t\right) n_y + \left(w_{ref} - z_t\right) n_z \tag{25}$$

The energy at the surface can be calculated through:

$$E_{body} = \frac{p_{body}}{\gamma - 1} + \frac{1}{2} \rho_{body} \left(u_{body}^2 + v_{body}^2 + w_{body}^2\right) \tag{26}$$

The characteristic boundary conditions are applied on all far field boundaries in Beggar. For supersonic flow, all waves run downstream and the boundary values are always easily specified. In subsonic flow, there are four downstream and one upstream running waves [33]. The missing information needed to specify the values of the conserved variables on the boundary comes either from the numerical solution inside the computational domain or some specified physical value [15]. Because this differencing direction is important to the accuracy of the flow at the boundary, it is equally important to include the motion of the mesh when determining the direction of the far field flow. This is done by using the contravariant velocity shown in Equation 25.

2.2 Overset Grids in Beggar

Beggar is strictly a three-dimensional structured grid solver. A structured grid is a rectangular arrangement of x, y, and z grid points (vertexes) that define a three-dimensional curvilinear coordinate system of i, j, and k points in computational space. Structured grids are advantageous because of the way the physical domain easily maps into the computational domain. For viscous problems, structured grids also make it easier to model the boundary layer efficiently. Unfortunately, a major disadvantage of structured grids is the process of grid generation for complex geometries, which any store separation or moving body problem naturally contains.

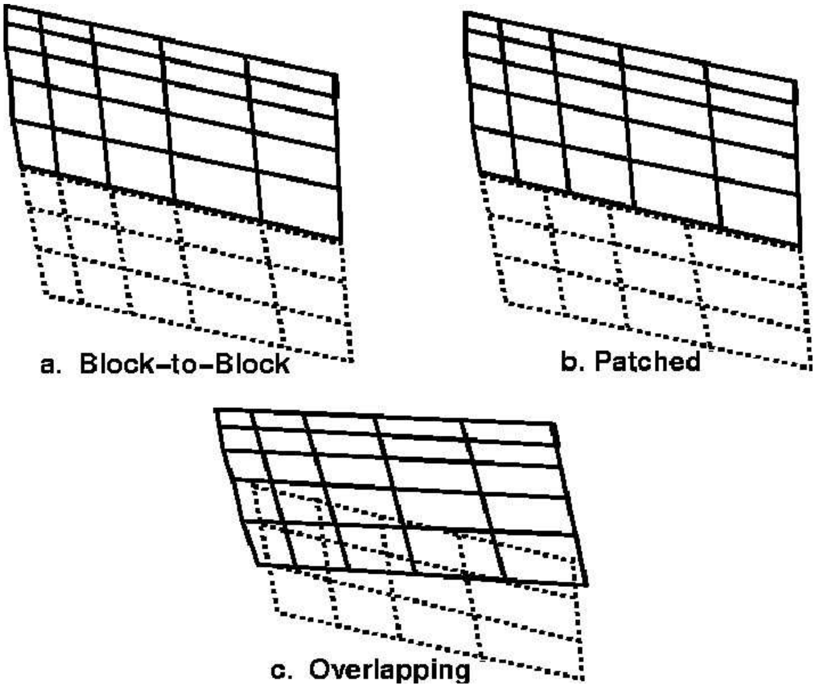

Figure 3: Block-to-Block, Patched, and Overlapping Grid Communications [1]

Beggar uses block-to-block, patched, and overlapping grids to simplify the treatment of these complex geometries, as shown in Figure 3. This means each grid needs to be generated only once and may be done independently of the other grids. Care must still be taken to ensure sufficient overlap between grids to enable grid-to-grid communication and minimize areas of significant grid cell size differences [24]. Points with no interpolation stencil (orphans) should also be minimized to maintain accuracy of the solution.

2.2.1 Grid Hierarchy. It is important to understand the grid hierarchy structure used in Beggar to better understand grid assembly and communication. The three main structures are the superblock, dynamic group, and super dynamic group [26].

The superblock is the primary grid structure in the Beggar grid assembly process. A superblock may contain one grid or multiple grids grouped together with block-to-block or patched communication. Computational efficiency can be increased greatly by using multiple grids grouped together into one superblock. The superblock construct is beneficial because the ease of domain decomposition allows many configurations to be built quickly. In fact, the use of superblocks is required when there is relative motion between two bodies. Building a superblock out of grids that communicate with only block-to-block or patched boundaries may be difficult, though. It is left to the user to decide what approach to take on a particular problem. Figure 4 shows a superblock consisting of two grids with block-to-block boundaries, which is used as the background mesh in this research.

Often multiple superblocks are used in a problem. The largest superblock, commonly called the background mesh, is used to simulate the flow far away from the bodies of interest. The other superblocks are contained inside the background mesh to model the body of interest.

A dynamic group consists of one or more superblocks that are moving as a single entity relative to the inertial reference frame. A dynamic group is also used

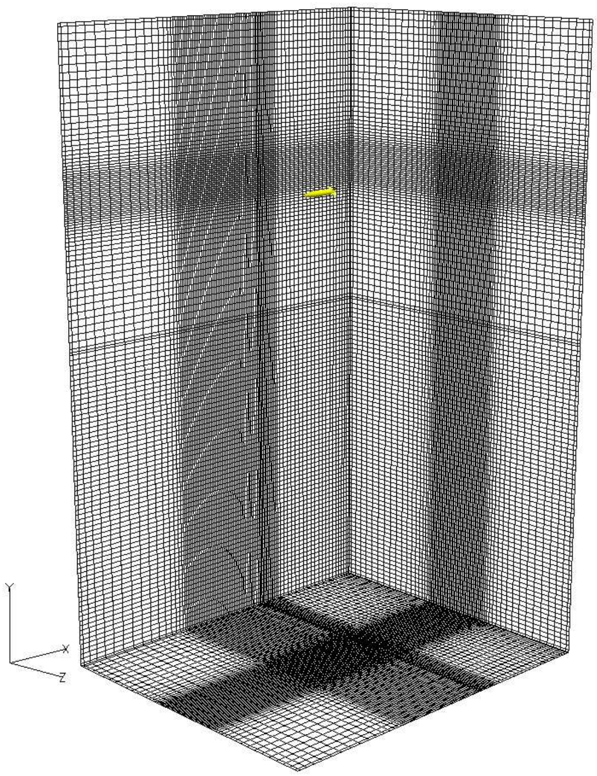

Figure 4: Tall Background Mesh surrounding MK-84 AIR body

to specify values needed for the dynamic problem, such as mass, moments of inertia, reference lengths, or output specifications. These specifications are contained in the dynamic specification contained in the Beggar input file. A single force specification identifies which surfaces to record the coefficients of forces and moments calculated by the (6+)DOF as well as file output options.

A super dynamic group is used in a multi-body problem such as a store with attached moving components. The super dynamic group contains the separate dynamic groups and dynamic specifications of each component, as well as a dynamic specification for the control of the super dynamic group.

2.2.2 Grid Communication. The three types of grid-to-grid boundaries are shown in Figure 3. The first type, block-to-block, occurs when two grids meet along a boundary with the alignment of coordinate lines across grids. The two cell faces on the adjoining boundary must be identical, sharing exact vertex coordinates. In this case, the two layers of phantom cells that Beggar uses to apply boundary conditions will be filled with the dependent variables from the first two layers of cells in the adjoining grid. This is the most correct and efficient grid-to-grid boundary. It is important to note that while a singularity (a collapsed boundary face) also meets these conditions, Beggar treats a singularity like a patched boundary condition.

Patched boundaries are similar to block-to-block except the grid faces do not have to exactly match. The location of the two layers of phantom cells are extrapolated out from the boundary, and the values therein are interpolated from the cells inside the adjacent grid.

All communication within a superblock occurs either through blocked or patched boundary conditions. All communication between superblocks is accomplished through overlapping grid communication [1]. In this case, grid interpolation is required to exchange information between superblocks, which is accomplished with the overset, or Chimera, grid assembly process. This process finds any region of one grid which lies within another grid or solid region and then removes that region from the simulation [24]. The automated Chimera grid assembly procedure in Beggar follows several steps in this process [17]:

1. Build data structures with grid boundary points.
2. Identify symmetry plane and singularities.
3. Establish grid connections for the point-matched boundaries.
4. Insert the grid boundary facets into the octree.
5. Classify the octree nodes as inside, outside, or on the boundary of the grid.
6. Establish grid connections for the non-point-matched boundaries.
7. Identify far field boundary faces that weren't specified in the input.
8. Cut holes.
9. Mark fringes, boundaries, and locate interpolation source stencils.

Beggar fills interior cell values of one grid through interpolation of the eight surrounding cell centers belonging to the other grid. This value is also weighted according to the position of the interior cell with respect to the interpolation field. An edge of a superblock will commonly define a surface of a solid body, the inside of which is outside the solution domain. In this case, the cells inside the body are removed from the solution domain (referred to as hole cutting) and the interpolation is determined for the cells surrounding the hole (fringe points).

For a dynamic problem, this process must occur every iteration as the body moves. Furthermore, this motion may cause grid cells which were previously inside the solid body to rejoin the computational domain. This may happen in one of two ways. If the grid cell is partially uncovered in a single time step, it is treated as a fringe point and is interpolated. A grid cell which is completely uncovered in a single time step does not normally happen, so Beggar does not address this.

2.3 Six Degree-of-Freedom Model

Beggar performs four steps to determine a store's motion in the flow. The first step is grid assembly, which establishes communication between grids, discussed previously. Next, the governing equations as presented in Section 2.1 are solved across the solution domain. The pressures along the body of interest are then integrated to determine the forces and moments on the body. Finally, the body is moved according to the solution of dynamic equations of motion obtained by the (6+)DOF solver.

The original Beggar 6DOF solver is improved upon greatly in the current (6+)DOF solver [27]. The (6+)DOF solver allows each rigid store to have store moving components such as fins that are modeled separately. These SMCs may rotate independently of the store main body about some axis that is fixed to the SMB. They may also follow some prescribed motion relative to the SMB around this axis. In this study, the MK-84 analyzed does not have movable fins, and thus is modeled as one rigid body. The generic store body used in this research has four moving fins associated with it, controlled through a prescribed motion set by the user.

Beggar incorporates this (6+)DOF model into the dynamic solver which allows bodies and components to move rigidly through the flow. The coupling of the CFD solver and the Newton-Euler equations for motion allows these bodies to rotate and translate freely in response to the forces and moments found by the CFD solver [21].

2.3.1 Beggar Coordinate Systems. Every superblock used in Beggar has a local coordinate system associated with it, which is unique to that grid. To understand how an object can move in response to the (6+)DOF, it is important to have an understanding of the coordinate systems internal to Beggar.

For this discussion, it is assumed that the standard CFD orthogonal coordinate system is used in all the grids, as seen in Figure 5. That is, the positive x-axis points downstream, the positive y-axis points upwards, and the positive z-axis points out the left wing (from the pilot's perspective). As is customary, the grids used in this research have been generated with the $\langle 0, 0, 0 \rangle$ location at the nose of the store and the x-axis running along the body from nose to tail. Beggar also assumes the customary positive flow direction in the positive x-axis direction.

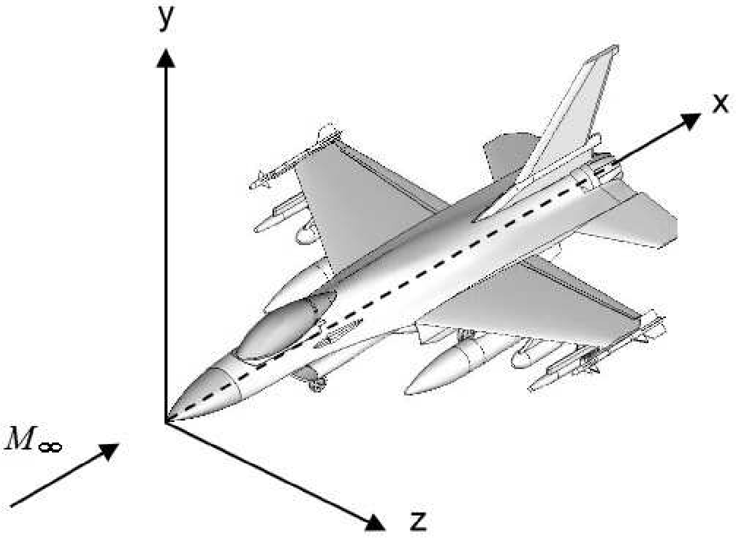

Figure 5: Standard CFD Coordinate System

When the grids are initially read into Beggar, all grids are assumed to have been generated in the fixed, global coordinate system, and the grid coordinates are used exactly as read from the grid files. An object may be placed in a certain initial orientation in the flow by rotating, translating, or scaling that object's grid to the desired position. Any such transformation made to the initial grid locations can be handled in one of two ways in the initial grid assembly.

If these transformations are specified in the superblock portion of the input, the grid coordinates are immediately modified and the new coordinates are stored [1]. Otherwise, if no initial transformation is specified, the grid coordinates are left unchanged and the initial local coordinate system corresponds to the global coordinate system.

However, if the initial transformations are specified in the dynamic spec, the local grid coordinates are not changed [1]. Instead, an initial transformation matrix is created which positions the local grid relative to the global coordinate system. The advantage of this approach comes from the way the moments and products of inertia are defined. These are defined relative to the "mass centered" coordinate axes, which are aligned with the local coordinate axes, centered at the center of gravity. Because of this, they remain constant as the store moves. If these were defined relative to the global axes, they would change with the motion of the store. So, if the transformations were placed in the superblock scope, the local coordinates would be modified and may no longer align with the "mass-centered" coordinate axes. The local coordinate system would still move with the store, but the moments and products of inertia may no longer be exactly equal to the true properties.

A multi-body problem has multiple coordinate systems associated with each component. For a fin and store combination, there are the original two coordinate systems fixed to each body plus two additional coordinate systems. The first is fixed to the fin at the specified point of rotation with its axes aligned with the fin's axes. The second additional coordinate system is fixed to the store with its axes aligned with the second fin coordinate system. These two coordinate systems are initially coplanar and coincident, but this changes as the fin rotates about the axis of rotation. Each additional fin in a multi-body problem has its own unique coplanar coordinate system.

2.3.2 Transformations in Beggar. As previously stated, each grid has its own local coordinate system which remains unchanged after initial grid assembly. Each of these local grids also has a transformation matrix associated with it which transforms a point in the local coordinate system to a point in the global coordinate system, and vice-versa. In the multi-body problem, there are additional transformation matrices between every component and every other component. When a body is moved in response to the equations of motion, these transformation matrices are changed, while the local coordinates continue to remain unchanged. These matrices are used when finding the interpolation stencils from one grid to another overlapping grid [25]. What follows here is a fundamental introduction to transformation matrices and how they are used in Beggar.

Any transformation may be divided into its rotational components and translational components. This is exactly how Beggar carries the transformations internally. One variable contains the components of the 3x3 rotational matrix and another variable contains the three components of the translation. The concept of a translation is simple enough: a vector containing the amount of change in the three directions ($[T_{x,y,z}]$) is added to the original position vector to obtain the new positions (x', y', z') as shown here.

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} + [T_{x,y,z}] \tag{27}$$

The subject of rotations is slightly more complex but follows the same principles. In this discussion, the triplet of Euler angles (α, β, γ) are used to describe the angles of rotation about the x, y, z axes, as shown in Figure 6. In addition, the common right-handed convention is used, so that all rotations are positive in the counter-clockwise direction.

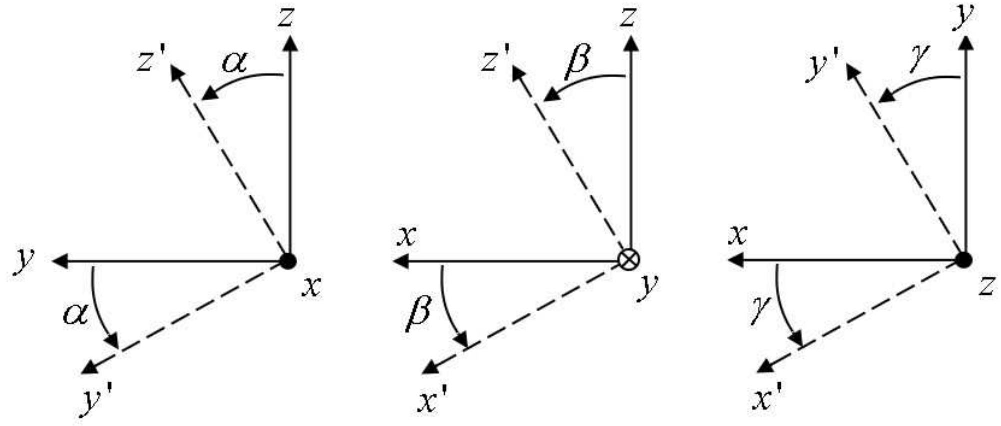

Figure 6: Euler Rotation Angles

Any three-dimensional rotation can be divided into three separate rotations about each axis. A rotation about the x-axis using the angle α is accomplished with:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \tag{28}$$

and results in the rotation seen in Figure 7.

Similarly, rotations can be accomplished about the other two axes with the following rotation matrices:

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, R_z(\gamma) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{29}$$

These three rotation matrices may be combined into one that describes the rotation about all three axes and angles:

$$R_{x,y,z}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos(\alpha)\cos(\beta) - \sin(\gamma)\sin(\alpha)\sin(\beta) & -\sin(\gamma)\cos(\alpha) & \cos(\gamma)\sin(\beta) + \sin(\alpha)\sin(\gamma)\cos(\beta) \\ \sin(\gamma)\cos(\beta) + \cos(\gamma)\sin(\alpha)\sin(\beta) & \cos(\gamma)\cos(\alpha) & \sin(\beta)\sin(\gamma) - \cos(\gamma)\sin(\alpha)\cos(\beta) \\ -\sin(\beta)\cos(\alpha) & \sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix} \tag{30}$$

Now, the coordinates of a point after a three dimensional rotation may be found with this single matrix:

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} [R_{x,y,z}] \tag{31}$$

A full transformation, with rotation and translation, can now be accomplished through:

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} [R_{x,y,z}] + [T_{x,y,z}] \tag{32}$$

The matrices $[R_{x,y,z}]$ and $[T_{x,y,z}]$ are the transformation matrices contained inside Beggar, which define the location and orientation of every coordinate system with respect to every other coordinate system contained in the problem, including the inertial reference frame. When a grid moves in response to the (6+)DOF, it is these matrices that are updated to reflect that motion.

Because of the difficulty associated with rotating an object about an arbitrary axis, Beggar performs four steps to rotate a dynamic group, shown in Figure 8:

1. Translate to $\langle 0, 0, 0 \rangle$ in global coordinates by subtracting $[CG_{x,y,z}]$

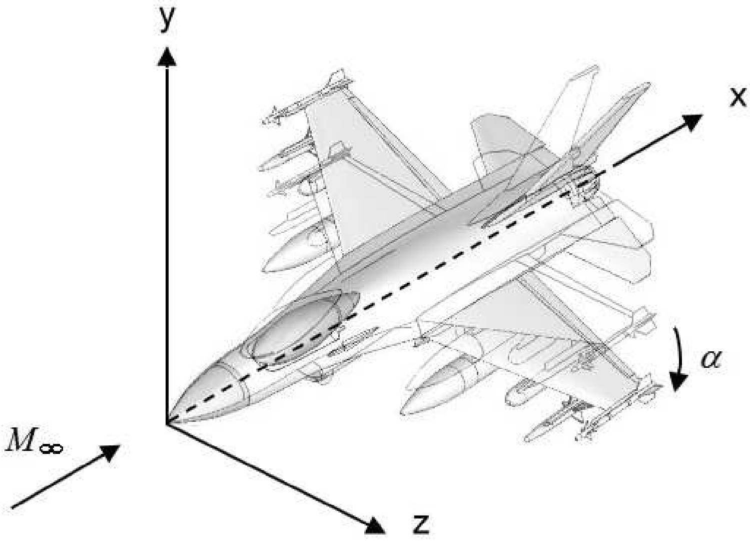

Figure 7: Rotation about the x axis

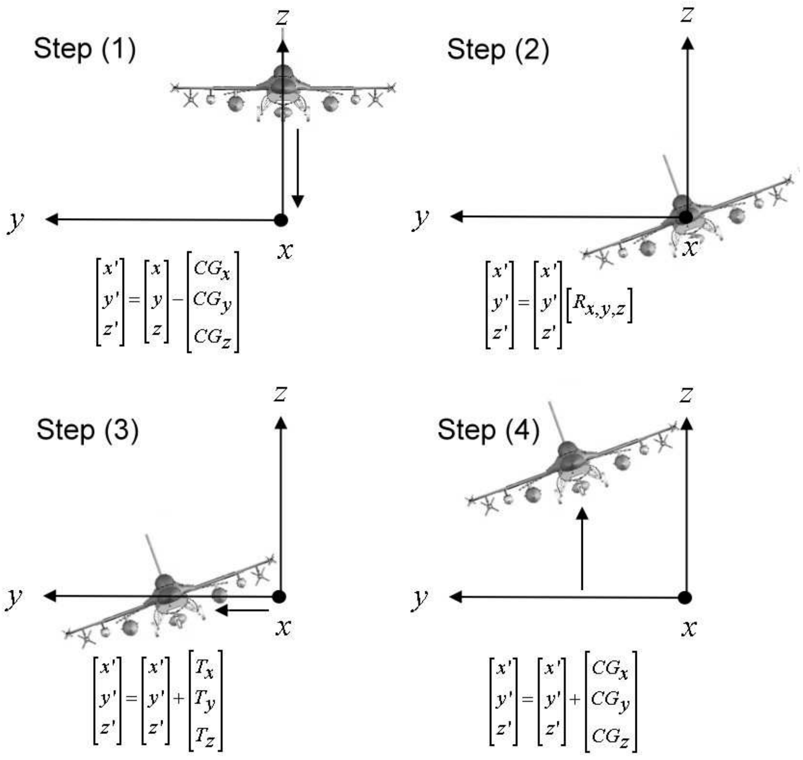
Step (1)
z
y
x
$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\begin{bmatrix}x\\y\\z\end{bmatrix}-\begin{bmatrix}CG_x\\CG_y\\CG_z\end{bmatrix}$
Step (2)
z
y
x
$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\begin{bmatrix}x'\\y'\\z'\end{bmatrix}\left[R_{x,y,z}\right]$
Step (3)
z
y
x
$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\begin{bmatrix}x'\\y'\\z'\end{bmatrix}+\begin{bmatrix}T_x\\T_y\\T_z\end{bmatrix}$
Step (4)
z
y
x
$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\begin{bmatrix}x'\\y'\\z'\end{bmatrix}+\begin{bmatrix}CG_x\\CG_y\\CG_z\end{bmatrix}$

Figure 8: Rotation/Translation Process in Beggar

2. Rotate in response to (6+)DOF

3. Translate in response to (6+)DOF

4. Reverse first translation by adding $[CG_{x,y,z}]$

where $[CG_{x,y,z}]$ is the matrix containing the x, y, z location of the center of gravity of the dynamic group. Mathematically, this can be represented as:

$$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\left(\begin{bmatrix}x\\y\\z\end{bmatrix}-\begin{bmatrix}CG_x\\CG_y\\CG_z\end{bmatrix}\right)[R_{x,y,z}]+\begin{bmatrix}T_x\\T_y\\T_z\end{bmatrix}+\begin{bmatrix}CG_x\\CG_y\\CG_z\end{bmatrix} \tag{33}$$

The subject of the transformations of store moving components (SMCs) is slightly more complex but follows the same basic principles. Each SMC has two coordinate systems associated with it: the fin coordinate system and the coplanar coordinate system that is aligned with the axis of rotation. The second coordinate

Table 2: Coordinate System Abbreviations

Abbr:	**Coordinate System**
F	Fin
FCO	Fin Coplanar
$FCOI$	Fin Coplanar Initial
G	Global
S	Store

system is coplanar with another coordinate system attached to the store main body (SMB) with its axis also aligned with the fin's axis of rotation. These last two coordinate systems are initially coincident on the axis of rotation, with the angle ϕ_{F2FCOI} between them changing as the fin rotates.

As the dynamic group moves, the fin is pinned at its hinge point adjacent to the SMB. Therefore, any rotations or translations of the SMB must also be reflected in the transformations of the fin to know the position of the fin relative to the store at all times. As discussed earlier, multiple transformation matrices may be combined through matrix multiplication to obtain a single transformation matrix governing the position of an object. For the fin, this is accomplished through a long chain of transformations which connect the transformation matrix between the store and global reference frame (A_{S2G}) to the transformation matrix between the fin and fin coplanar coordinate systems (A_{F2FCO}). This relationship is shown in detail here.

The rotational transformation matrix from the fin to the global reference frame is found through Equation 34, where $[A]$ refers to the rotational transformation matrix between the two coordinate systems listed in the subscripts. A list of the different coordinate systems used here is given in Table 2.

$$[A_{F2G}] = \underbrace{\left[[A_{S2G}] \underbrace{\left[[A_{S2FCOI}]^{-1} [A_{FCO2FCOI}] \right]}_{A_{FCO2S}} \right]}_{A_{FCO2G}} [A_{F2FCO}] \tag{34}$$

Then the fin to store transformation matrix ($[A_{F2S}]$) can be found through :

$$[A_{F2S}] = [A_{G2S}]\,[A_{F2G}] \tag{35}$$

Finally, the translation of the fin ($[T_{FIN}]$) is found from the translations of the SMB and the position vector from the fin to global coordinate system ($[\rho_{F2G}]$).

$$[T_{FIN}] = [T_{SMB}] + [T_{CG_{SMB}}] + \underbrace{[A_{S2G}]\,\underbrace{([[A_{F2S}]\,[\rho_{H2F}]]\,[\rho_{H2S}])}_{\rho_{S2F}}}_{\rho_{F2G}} \tag{36}$$

where $[\rho_{H2F}]$ and $[\rho_{H2S}]$ are the position vectors from the hinge to fin and hinge to store, respectively. The translations of the SMB are separated into two parts; one part is the translations associated with the center of gravity ($[T_{CG_{SMB}}]$), and the other part is the non-zero component of translations that result from the rotational process in Equation 33 and Figure 8.

2.3.3 Dynamic Equations of Motion. The equations used in the Beggar implementation of the (6+)DOF are presented here. Newton's Second Law of motion can be applied to the store main body through the form [27]:

$$F^{s}_{appl} + \sum_{j=1}^{J} F^{js}_{const} = m^{s}\frac{d^{I}}{dt}\left(v^{s}\right) \tag{37}$$

The superscripts s, j and c denote the centers of mass of the store main body, the j^{th} SMC (out of J total moving components) and the entire store, respectively. The quantity F^{s}_{appl} is the applied forces experienced by the SMB, which consists of aerodynamic and gravitational forces, as well as ejector forces. The quantity F^{js}_{const} is the constraint force exerted by the store moving components on the store main body, keeping in mind that equal and opposite forces are exerted by the SMB on the SMCs. These constraint forces cause the equations of motion to be coupled. The time derivative $\frac{d^{I}}{dt}$ is the time derivative as observed from the inertial reference frame.

The equation governing the angular velocity of the store main body is shown here. This relationship is derived from the law that says the rate of change of the angular momentum of a store component is equal to the moment of the forces experienced by that component, all about the center of mass (CM) of that component [27].

$$N^s_{s(appl)} + \sum_{j=1}^{J} N^{js}_{s(constr)} = \frac{d^I}{dt}\left(I^s_s \omega^s\right) \tag{38}$$

where $N^s_{s(appl)}$ is the moment of applied forces on the SMB and $N^{js}_{s(constr)}$ is the moment of the constraint force exerted by the j^{th} store moving component on the store main body about the center of mass. The SMB inertia tensor about the center of mass is denoted with I^s_s and the angular velocity is denoted with ω. The equation governing the angular velocity of the SMCs is the same, except with the appropriate subscripts denoting the moving components:

$$N^j_{j(appl)} + \sum_{j=1}^{J} N^{sj}_{j(constr)} = \frac{d^I}{dt}\left(I^j_j \omega^j\right) \tag{39}$$

The constraint forces of the SMCs on the SMB are now rewritten in terms of the applied forces and dependent variables [27].

$$F^{sj}_{constr} = m^j a_j - F^j_{appl} \tag{40}$$

where a_j is the acceleration of the center of mass of the j^{th} moving component. The constraint moment on the SMB by the j^{th} moving component is:

$$N^{js}_{s(constr)} = -\left(N^{sj}_{j(constr)} + \rho_{sj} \times F^{sj}_{constr}\right) \tag{41}$$

where ρ_{sj} is the is the position vector of the SMC center of mass relative to the SMB center of mass. The SMB also applies a constraint force to the j^{th} SMC, which can be expressed as:

$$N^{sj}_{s(constr)} = \left(N^{sj}_{h_j(constr)} + \rho_{jh_j} \times F^{sj}_{constr}\right) \tag{42}$$

where h_j is the hinge point between two components.

For the single body problem, the equations of motion simplify to:

$$F^{s}_{appl} = m^{s}\frac{d^I}{dt}\left(v^s\right) \tag{43}$$

$$N^{s}_{s(appl)} = \frac{d^I}{dt}\left(I^s_s\omega^s\right) \tag{44}$$

The above equations govern the velocity and angular velocity of the store main body. The trajectory of the SMB center of mass can be computed through:

$$\frac{d^I}{dt}x_s = v_s \tag{45}$$

where the position vector of the SMB center of mass is contained in x_s.

The orientation of the store must now be obtained. This is done using the coordinate systems and transformation matrices discussed in Section 2.3.2. The i, s and j coordinate systems used here are fixed to the inertial reference frame, SMB, and the j^{th} SMC, respectively. To maximize computational efficiency, Beggar uses quaternions internally to define the orientation and state of the SMB [25]. Only a brief overview of quaternions is provided here. A fundamental understanding of quaternion mathematics may be gained from [3, 13, 28].

Quaternions are scaled vectors specifically used for Eigen-axis rotations of coordinate systems [34], which are advantageous because operations on them can be computed more efficiently. In addition, they have no pitch singularity as the Euler angles have [23]. Each quaternion is a hypercomplex number consisting of one real and three imaginary parts through the form:

$$q = q_s + q_x + q_y + q_z \tag{46}$$

This may also be interpreted as a scalar component and a vector component [28], where the scalar part of the quaternion is defined by q_s and the rest of the quaternion is the vector component q_v. When considering a rotation angle θ about some axis of rotation, the scalar part is equal to $cos(\theta/2)$ while the vector part is a unit vector along that axis multiplied by $sin(\theta/2)$ [27]. An extra constraint is required with quaternions [23]: $q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1$.

The quaternion elements may be found at any point in time through the two equations:

$$\frac{d}{dt} q_s = -\frac{1}{2} q_v \omega^s \tag{47}$$

$$\frac{d^I}{dt} q_v = \frac{1}{2} q_s \omega^s + \frac{1}{2} \omega^s \times q_v \tag{48}$$

A transformation matrix defining the orientation of the SMB with respect to the inertial reference frame may now be found at any point in time through the quaternion elements as shown in Equation 49:

$$T^{si} = \begin{bmatrix} 2\left(q_y^2 + q_z^2\right) - 1 & 2\left(q_{xy} - q_{zs}\right) & 2\left(q_{xz} + q_{ys}\right) \\ 2\left(q_{xy} + q_{zs}\right) & 2\left(q_x^2 + q_z^2\right) - 1 & 2\left(q_{yz} - q_{xs}\right) \\ 2\left(q_{xz} - q_{ys}\right) & 2\left(q_{yz} + q_{xs}\right) & 2\left(q_x^2 + q_y^2\right) - 1 \end{bmatrix} \tag{49}$$

where, for example, q_{xy} is the multiplication of the two quaternions elements: $q_{xy} = q_x q_y$.

To show that this is equal to transformation matrices obtained in Section 2.3.2, we can simulate a rotation about the x-axis, setting the quaternion $q = cos(\theta/2) + sin(\theta/2) + 0 + 0$. This results in the transformation matrix:

$$T^{si} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2\left(\cos\left(\theta/2\right)^2\right) - 1 & 2\left(-\sin\left(\theta/2\right)\cos\left(\theta/2\right)\right) \\ 0 & 2\left(\sin\left(\theta/2\right)\cos\left(\theta/2\right)\right) & 2\left(\cos\left(\theta/2\right)^2\right) - 1 \end{bmatrix} \tag{50}$$

which, after some trigonometric simplification, results in Equation 28.

2.3.4 Equation of Motion Solution Method. The applied forces and moments acting on the SMB are a required input to the (6+)DOF solver. The solution of the flow solver provides the pressure and viscous stress distributions along the body surfaces [26]. Numerical integration after every iteration provides the forces and moments acting on the body [25].

A system of linear, ordinary differential equations may be set up with the dynamic equations of motion, shown in Equations 43 to 45 and 47, 48. This system is solved in Beggar using a fourth order Runge-Kutta scheme [25]. The fourth order Runge-Kutta scheme is defined as [8]:

$$y_{n+1} = y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{51}$$

where h is the time step and the coefficients k_1, k_2, k_3, and k_4 are shown here for completeness:

$$\begin{aligned} k_1 &= f\left(x_n, y_n\right) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{hk_1}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{hk_2}{2}\right) \\ k_4 &= f\left(x_n + h, y_n + hk_3\right) \end{aligned} \tag{52}$$

Although the aerodynamic forces are assumed constant over a time step, the gravitational force changes because it is a function of position in the local coordinate system. Therefore, after every Runge-Kutta step, the SMB state vector must be updated, and the transformation matrices, forces and moments are recalculated [2].

2.4 Current Approach

The current research seeks to remove the translations of the store through the background mesh and reflect the translational velocities of the store as changes in the grid speeds of the background mesh through the convective flux vector. This must be done without affecting the flow solution, forces and moments, or telemetry data of the Beggar solution.

As seen in Section 2.3.2, there are four steps that are needed to move the store in response to the forces and moments, and three of these steps are translations. In order to correctly rotate the store, the first and last translations must be preserved because of the way any rotation must be performed about the origin. In other words, if all the translations associated with a dynamic group are continually set to zero, the transformation matrix will be incorrect. Instead of removing all translations, the correct translation to remove is the third step in that process, shown in Figure 7. If that translation, which is done in response to the forces acting on the store, is removed when Beggar is building the transformation matrices, then the dynamic group will effectively be pinned at whatever location it is initially placed in the background mesh. This will allow the non-zero translations of Steps 2 and 4 of the rotation process to remain in the store's transformation matrix, as they must do for correct positioning of the store.

The removal of the inertial reference frame presents some problems when determining the accurate location of the store. The effect of these modifications is that the background mesh travels with the store as it falls through the air. Consequently, the location of the store is not able to be measured from this new reference frame. The changes in the store's location and orientation are instead measured against the saved, unmodified transformation matrix, which is the original inertial reference frame. This allows the store's position and orientation in extended free flight to be accurately calculated throughout the solution.

Now the translational velocities of the store may be applied to the grid speed of the background mesh. This is done through the convective flux vector ($\vec{F}_c$) shown in Equation 3. These changes will allow successful grid assembly to continue throughout the simulation while preserving the forces and moments acting on the store. Effectively, the simulation may now proceed for an indefinite amount of time.

III. Methodology

Code modifications to Beggar are the first step required in this research. These modifications are made in phases, with the appropriate testing also being completed in phases. The first phase of modification entails applying some arbitrary mesh motion to the background mesh and confirming the successful application of that mesh motion. In addition, the flow solution output is corrected to include the mesh motion in this phase. This is confirmed through the use of a simple supersonic compression ramp. The second phase involves the pinning of the store body in the background mesh and the application of the store velocities to the background mesh motion. The most extensive testing is done after this stage, with comparisons of the MK-84 AIR store forces and moments, velocities, and telemetry data being conducted. Finally, modifications are made to enable the use of store moving components on the pinned store main body. A test case of a pitching generic store body is then conducted to demonstrate the usefulness of this research to the analysis of dynamic flight characteristics.

3.1 Beggar Modifications

The latest version of Beggar (version 114j) was obtained from the Air Force Seek Eagle Office and used as the starting point for all modifications required for this research. Some enhancements to this version include improved parallel processing capabilities, an extended force output, and a new, two-stage Euler scheme for solving the dynamic equations of motion.

3.1.1 Mesh Motion. Prior to applying a velocity to the background mesh, the translational velocity of the store is obtained. Every dynamic group created in Beggar carries the state vectors of its components in an internal data structure. This information is available to every processor, which is beneficial considering the store grids and background grids may not necessarily exist on the same processor in a parallel run. In the main iteration loop of the flow solver, prior to looping over the superblocks, the code is modified to loop over the dynamic groups present and extract the u, v, and w velocities from the store state vector. Three new variables are created to hold these values. When the solver enters the main superblock loop, the time metrics (the local grid speeds) are calculated on the first dt iteration of each time step. Normally, the background mesh is inertial and does not have any grid speed associated with it. This code is modified to apply the newly found store velocity to the background mesh. This is done by passing the velocity variables into a redesigned metrics routine. Instead of completely transforming the spacial metrics and calculating the grid speeds like the normal metrics routine, this new routine simply accepts the passed values of the store u, v, and w as the grid speeds and applies them directly to the background mesh, whose identifying grid number is also passed into this routine. The solver then continues as normal and solves the governing equations with the applied mesh motion in the convective flux vectors as shown in Equation 3.

3.1.2 Flow Visualization. Because the mesh contains some portion of the motion in the system, the conserved variable vector $\vec{Q}$ no longer represents the total flow velocity. However, the $\vec{Q}$ vector is all that is normally written to the solution file. Thus, some adjustments are needed to correctly output the total flow velocity to the file. Because of Beggar's ability to output solution files at any iteration interval, this correction cannot be made to the $\vec{Q}$ vector itself. Instead, this correction is made to temporary variables that are created in Beggar's output routine and used to write the flow solution. The modifications to the temporary conserved variables are accomplished according to the following equations:

$$\begin{aligned} \rho &= \rho \\ \rho u &= \rho u - \rho x_t \\ \rho v &= \rho v - \rho y_t \\ \rho w &= \rho w - \rho z_t \end{aligned} \tag{53}$$

The total energy of the flow field is corrected through:

$$E_t = \frac{p}{\gamma - 1} + \frac{\rho}{2}\left((u - x_t)^2 + (v - y_t)^2 + (w - z_t)^2\right) \tag{54}$$

These new values are then written to the file, and accurately reflect the total flow properties that the store is experiencing.

3.1.3 Translation Eliminations. As discussed in Section 2.3.2, a matrix of transformation matrices exist in Beggar which are able to transform a point in one coordinate system to any other coordinate system contained in a problem, including the inertial reference frame. Achievement of the present goal hinges on the successful modification of these matrices. Complications arise because of the different ways these transformations are employed in Beggar. The main usage of transformation matrices is for grid interpolation. Some of the specific grid interpolation applications are: transforming a bounding box, hole cutting, grid communication, and stencil operations [2]. In order to simulate the store being pinned, the transformation matrices used in these areas should not include the translations of the SMB.

The second usage is when finding and applying forces and moments to the store. For example, the point that the moments are calculated about exists only in the local coordinate system of the dynamic group. The same is true of the dynamic group's center of gravity. The transformation matrix between the dynamic group and the global coordinate system is used to find these values in the global reference frame. These global values are then used to calculate additional moments about the CG in the global reference frame due to aerodynamic forces [2]. In addition, when applying ejector forces at a fixed point and direction in the global coordinate system, the current location of the dynamic group CG must first be found in global coordinates. The transformation matrix between the dynamic group and global coordinate system is also used here. In both of these cases, it is essential that the transformation matrix Beggar uses *does* contain the translations of the SMB. If it does not, the force coefficients and CG locations obtained will be incorrect and lead to a flawed solution.

To satisfy these two distinct uses of transformation matrices, two transformation matrices are built and carried in the modified code. The first is built *without* the translations of the SMB CG (Step 3 in Figure 8) included. The second transformation matrix is built *with* the translations of the SMB CG included. The function that transforms a value between two coordinate frames is duplicated; the second function is now modified to use the second transformation matrix. Now, two transformation matrices and transformation functions between each coordinate system exist. The first does not contain the translations of the store, and is used for grid assembly only. The second does include the store translations, and is used solely for the purposes of determining and applying forces and moments.

Special consideration is given to the multi-body problem when pinning the store in the background mesh. Any moving components must continue to rotate around the original axis of rotation relative to the store, which is no longer translating. Even though the store is not translating, a SMC may appear to translate because of its position relative to the SMB center of gravity. The rotations of the SMC are calculated through Equation 34 by using the relationship between the global, fin coplanar, and fin coordinate systems. The translations of the moving component are then calculated according to Equation 36. The translations of the SMB CG are plainly included in this calculation. However, in the pinned case, these translations *must not* be included in this equation to obtain the correct position of the SMCs relative to the store. This is implemented by removing the addition of the store main body translations ($[T_{SMB}]$) from this equation. The CG position of the SMB is still included through the $[T_{SMB.cg}]$ term, but this of course remains constant.

These adaptations to Beggar allow the store CG to remain pinned at its location in the background mesh while rotating freely about that point, yet allow the forces and moments to be calculated as if the store were really following its numerical trajectory. In addition, the important ability to simulate SMCs is preserved.

3.2 Supersonic Compression Ramp

The compression ramp, shown in Figure 9, was designed to provide a very simple means of confirming the correct application of mesh motion and flow visualization. This simple three dimensional grid has dimensions of 199 x 50 x 11 for a total of 109,450 cells. The turning angle of the ramp is 15 degrees. Because the turning angle and flow velocity are known, any supersonic flow over the ramp will produce an easily verifiable shock wave and Mach number behind the shock. In addition, any combination of flow and mesh motion over the ramp should produce a shock wave corresponding to the total flow velocity.

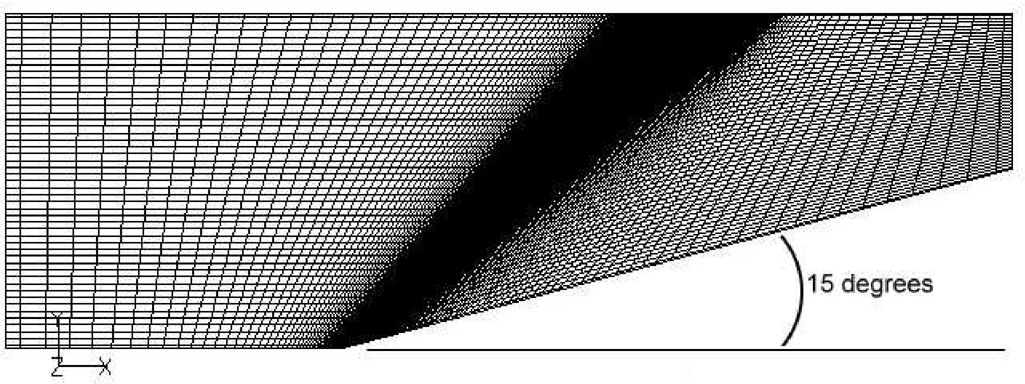

Figure 9: Supersonic Compression Ramp

If the mesh motion and flow visualization corrections are applied correctly, the correct shock angle and free stream Mach number will be seen in the post-processing visualization. If the combined Mach number is wrong, the shock angle will be wrong. If the shock angle is correct but the visualized Mach number in the flow solution is wrong, then the visualization corrections are miscalculated.

Three test cases were run using this compression ramp, shown in Table 3. All cases were run at sea level standard day conditions with a common total Mach number of 2.0. A CFL number of 5 was used to run the solution out for 1000 iterations. A reduction of four orders of magnitude in the L_2 norm is desired to indicate solution

Table 3: Compression Ramp Test Cases

	Mesh Mach Number	**Flow Mach Number**
Case 1	0	2.0
Case 2	0.5	1.5
Case 3	1.5	0.5

convergence. The 2.8 Ghz Pentium 4 Xeon® workstations with 4 GB of memory were used to obtain these single processor solutions.

The inviscid Euler solver is used with tangent boundary conditions on the bottom and side surfaces of the ramp. Far field boundary conditions are set on the inlet, outlet, and top surfaces. Beggar is inherently a 3D solver, but these boundary conditions provide a two-dimensional (2D) approximation of the flow over the ramp.

The first case is the reference case with no mesh motion, using the unmodified Beggar Code. The other two cases consist of varying combinations of applied mesh motion and flow velocity using the modified code. Being the only grid in this problem, the ramp is considered the background mesh and will demonstrate the modified code's ability to apply mesh motion to a background mesh.

3.3 MK-84 AIR

For this proof-of-concept research, a simple, ubiquitous store was desired in order to lighten the computational requirements and reduce restrictions on the presentation of numerical data. To meet these needs, the MK-84 low-drag, general purpose (GP) bomb, shown in Figure 10, was chosen. Often called a "dumb bomb", the MK-84 is an unguided munition used by the Air Force, Army, and Navy against a wide variety of targets [14]. For more precise bombing requirements, the Air Force commonly attaches a guidance kit to the Mk-84 to create laser guided bombs such as the GBU-10 Paveway II or GBU-24 Paveway III. Being the product of a 1950's development

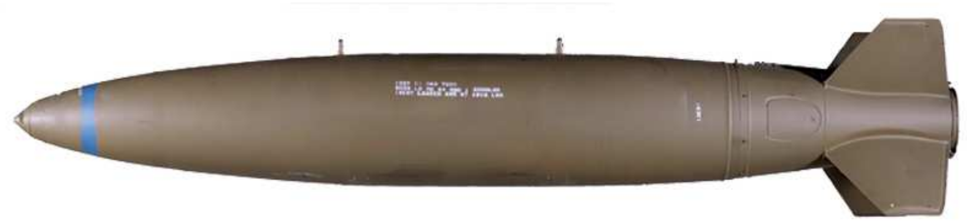

Figure 10: MK-84 General Purpose Bomb

effort to reduce aerodynamic drag on munitions, the MK-80 family of weapons is very streamlined, and the MK-84 is no exception.

The MK-84 AIR used in this research is a variant of the MK-84 modified with a BSU-50/B high drag tail assembly, used only by the Air Force. This tail assembly is made of a canister containing a ballute (combination of a balloon and parachute) retardation device. The ballute deploys from this canister to quickly slow the bomb by increasing drag, allowing the aircraft to escape the blast effects during high-speed, low-altitude bombing. For this research, the MK-84 AIR is modeled in the low-drag mode, with the ballute permanently stored. Some MK-84 specifications as given in *Jane's Air-Launched Weapons* are shown in Table 4.

3.3.1 Grid Generation. The computer model used to simulate the MK-84 AIR is shown in Figure 11. The MK-84 AIR geometry is modeled by four grids, each making up one quadrant of the store, shown in Figure 12. Initially, these grids were received from the AF Seek Eagle Office. They were then regenerated using Gridgen® to remove areas of high skew and uneven cell size. The radial distribution of cells was

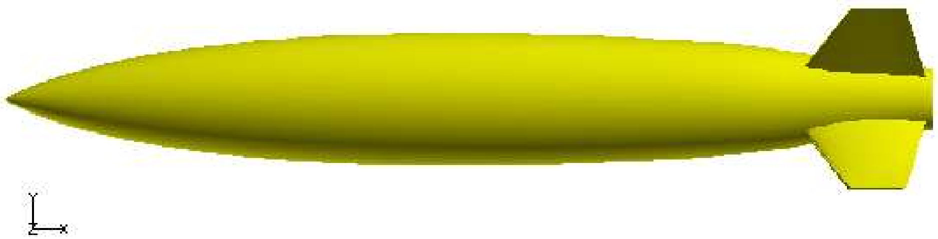

Figure 11: MK-84 AIR Computational Model

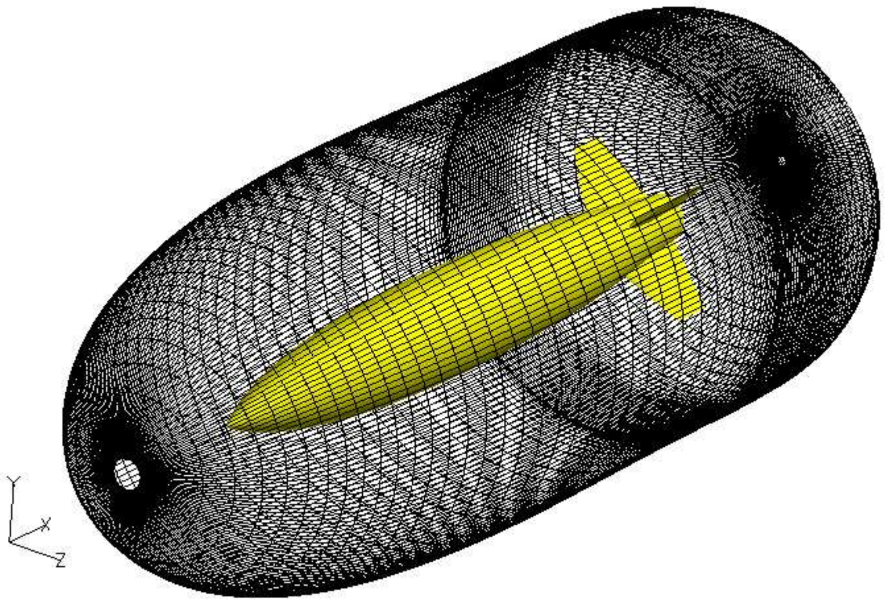

Figure 12: MK-84 AIR Superblock

also smoothed around the bomb body. Care was taken to preserve the asymmetric fins as defined by the original MK-84 grids.

Two background grids are generated and used. The first, shown in Figure 4, is designed to allow the bomb to fall through the mesh and thus is relatively ‘tall’. The second, shown in Figure 13, was designed for the pinned cases where the bomb does not translate through the mesh, and is only half the size of the first background mesh.

Table 4: MK-84 General Purpose Bomb Specifications

Primary Function:	2,000 lb GP bomb.
Weight:	2019 lbs.
Length:	10.75 ft.
Diameter:	18 in.
Tailspan:	25 in.
Explosive:	943 lb Tritonal

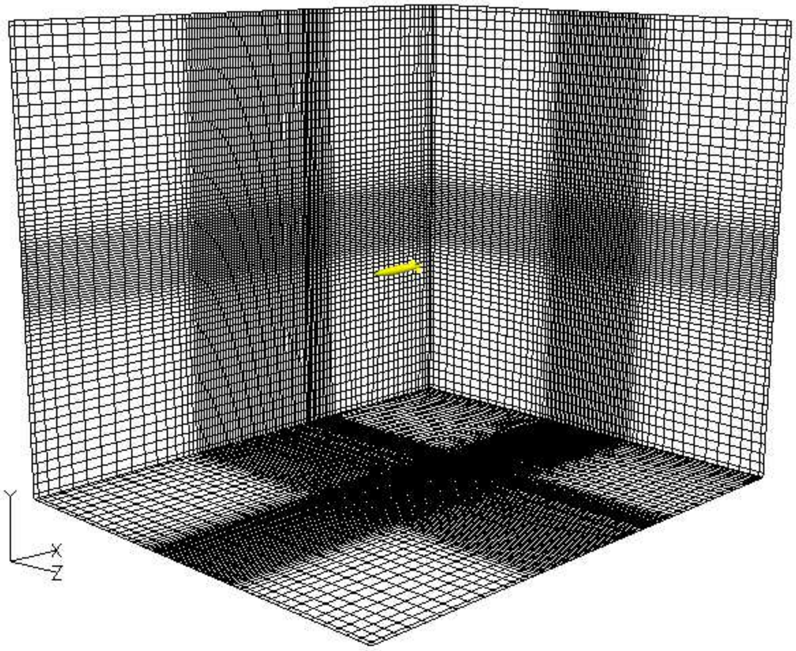

Figure 13: Small Background Mesh Surrounding MK-84 AIR Body

A simple cartesian grid consisting of relatively small cell sizes is used as an interface grid between the store grids and the background mesh. This grid is just large enough to surround the store grids and provide enough distance for grid interpolation with the larger cells of the background mesh.

3.3.2 Grid Dimensions. The dimensions of the grids used in this problem are shown in Table 5. With the four equally sized quadrants, there are 881,280 cells in the MK-84 AIR superblock. For the translating problem, the total number of grid cells in the computational domain is approximately 2.1 million; and for the pinned problem,

Table 5: MK-84 AIR Grid Dimensions

Grid	**Size**
MK-84 AIR Superblock	881,280 cells
Tall Background Mesh	970,140 cells
Small Background Mesh	456,950 cells
Cartesian Interface Mesh	564,376 cells

only 1.6 million cells are used. This shows the obvious computational advantage of removing the inertially fixed background mesh.

3.3.3 MK-84 AIR Testing. Before any solutions involving motion of the store are computed, the steady-state static solution is obtained. To keep the solution stable at all times, the time step is ramped from a small initial time step of 4.5 to a larger, 24.5 time step in 75 iterations. These time steps are given in non-dimensional quantities, and correspond to approximately 0.36 milliseconds and 1.96 milliseconds, respectively, depending on the speed of sound selected by the user. The Newton iterations are also ramped from 1 to 3 in this period. This ramping schedule is shown in Appendix B.3 and is chosen based on the recommendations of the Computational Aeromechanics Team. The 24.5 time step is then used to run the static solution out while monitoring the L_2 convergence norm and the forces and moments on the store body to determine when convergence is reached.

Dynamic testing begins by placing the store in the tall background mesh and allowing it to fall freely. The same time step of 24.5 is used from the beginning of this dynamic simulation. The length of the simulation is limited by the height of the background mesh, which corresponds to approximately 2.3 seconds of free fall for this background mesh. Care is taken to ensure the store does not approach the bottom of the background mesh too closely so that the quality of the solution is not corrupted. The unmodified Beggar code is used in this simulation to predict the motion of the store.

Next, the store is placed in the small background mesh shown in Figure 13. This solution is initialized from the static solution obtained in the same background mesh using the 24.5 non-dimensional time step. The fully modified Beggar code is used to pin the store in the background mesh and simulate the translations of the store through applied mesh motion as the solution progresses. Although the simulation could be run indefinitely, only the same solution time as the translating

Table 6: MK-84 AIR Test Cases

	Freestream Mach	**Altitude**
Case 1	0.6	20,000 ft
Case 2	0.9	10,000 ft

case is needed for this comparison. This process is repeated at the two test conditions shown in Table 6 to ensure the reproducibility of the results.

Finally, the strength of these modifications is shown by simulating the MK-84 AIR in free fall. Two test cases are run: the first is at Mach 0.6 at sea level standard conditions and the second case is Mach 0.9 at 10,000 ft. The store is released from the converged solution at these initial conditions and allowed to fall for an extended period of physical time. The trajectory, forces and moments, and velocities of the store are recorded through the free flight simulation. Beggar's ability to save solution files at specified intervals is also used to enable visualization of the entire free fall event. This extended simulation will model the MK-84 in free flight and reveal the dynamic characteristics of the store.

Beggar was used in conjunction with Parallel Virtual Machine (PVM) to obtain these solutions on multiple 2.2 Ghz AMD Opteron® processors. Three nodes were used for each run, where each node contains two of these processors sharing 4 GB of memory.

3.3.4 MK-84 AIR Numerical Validation. The numerical solution of the dynamic behavior of this store in free flight is difficult to validate. Difficulty arises because of the way the store is simply being "released" at altitude in the numerical simulation. Any flight test data obtained for the MK-84 trajectory and orientation would obviously involve some initial aircraft/store flow interactions as well as ejector forces on the store. These interactions would alter the store's behavior from the behavior of a clean release at altitude, thus making it difficult to draw any comparisons between flight test data and Beggar's extended numerical solution.

Wind tunnel data does exist for the validation of store separations, but is obtained using ejector forces in systems such as the Captive Trajectory System. In addition, any dynamic wind tunnel data is only available for a very short time period, usually on the order of tenths of seconds [10]. Wind tunnel static data is of little use in validating these dynamic simulations.

Because of these difficulties, the pinned numerical solution will be verified by simply comparing it to the translating numerical solution computed using the unmodified Beggar code. The solution from the pinned case should adequately match the solution from the translating case.

Different types of data are available to use in these comparisons between the translating and pinned methods. All the data presented in this research is purely numerical and is output every iteration of the flow solver. Much of the data herein is distinguished with the use of symbols. Any symbols do not represent a sampling rate, but are merely representative of that particular solution.

The Guidance & Control (G&C) telemetry data output by Beggar is used to compare the trajectory and orientation of the store throughout the simulation. This data is output every time step and contains the time, trajectory, and orientation of the store. The x, y, and z translations of the CG are given in feet from the origin of the coordinate system shown in Figure 14. All telemetry data presented in this research is in this coordinate system. The orientation angles of roll (ϕ), pitch (θ), and yaw (ψ) about the x, y, z axes, respectively, are given in degrees.

The forces and moments computed by the (6+)DOF are another way to determine if the store is experiencing similar effects via the pinned method. The forces and moments are only analyzed from the global reference frame (using the standard CFD coordinate system in Figure 5). Beggar outputs all forces and moments as coefficients through the following equations, where F and M refer to the force and moment and M_∞ refers to the free stream Mach number.

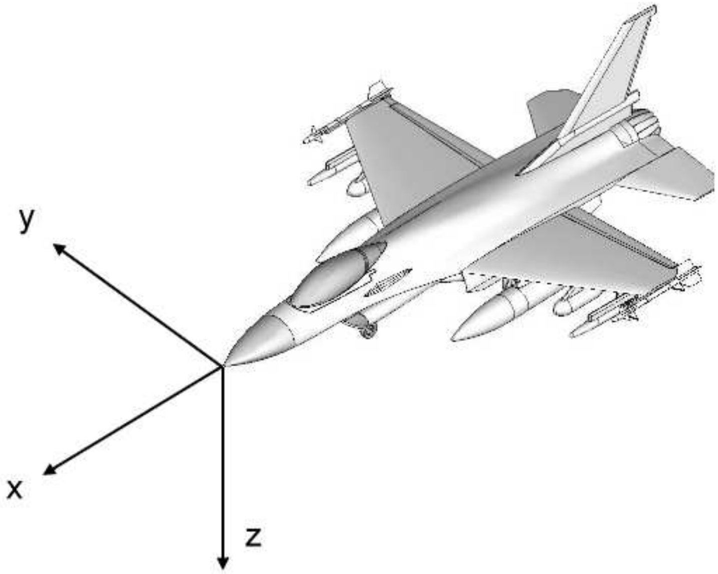

Figure 14: G&C Coordinate System

$$C_F = F\frac{2}{M_\infty^2 A_{ref}} \tag{55}$$

$$C_M = M\frac{2}{M_\infty^2 A_{ref} L_{ref}} \tag{56}$$

Finally, the translational and angular velocities of the store from the global reference frame are extracted from the solution history files and used as a means of comparing the two solutions. The translational velocity components (u, v, w) are output in feet per seconds, while the angular velocities $(\dot{\psi}, \dot{\theta}, \dot{\phi})$ are output in radians per second. These are also with respect to the standard CFD coordinate system.

With the capability of indefinite free flight, the store may reach its terminal velocity at some point in the simulation. The terminal velocity is the velocity at which the drag acting on the store balances the acceleration due to gravity and the store can no longer accelerate. This relationship is shown in Equation 57.

$$V_T = \sqrt{\frac{2W}{\rho C_d A}} \tag{57}$$

The drag is made up of two components: parasite drag and pressure drag. Because of the inviscid solver and tangent boundary conditions used in this research, there will be no parasite drag acting on the store and the base pressure on the store will be inaccurate. This means that at terminal velocity, the pressure drag will be the only force balancing the gravitational force. With this consideration in mind, the numerical terminal velocity is expected to be much higher than the published value of Mach 1.03 for the MK-84 AIR at sea level.

3.4 Generic Store with Moving Components

To demonstrate the effectiveness of these adaptations for the multi-body problem, a generic store body is used with four moving components (Figure 15. The SMCs are modeled as fins in the stable "X" configuration to control the pitching moment of the store.

3.4.1 Generic Store Grids. The generic store body superblock is composed of four grids using block-to-block communication. The fin superblocks are also modeled by four grids each, using block-to-block communication. The fin superblock is

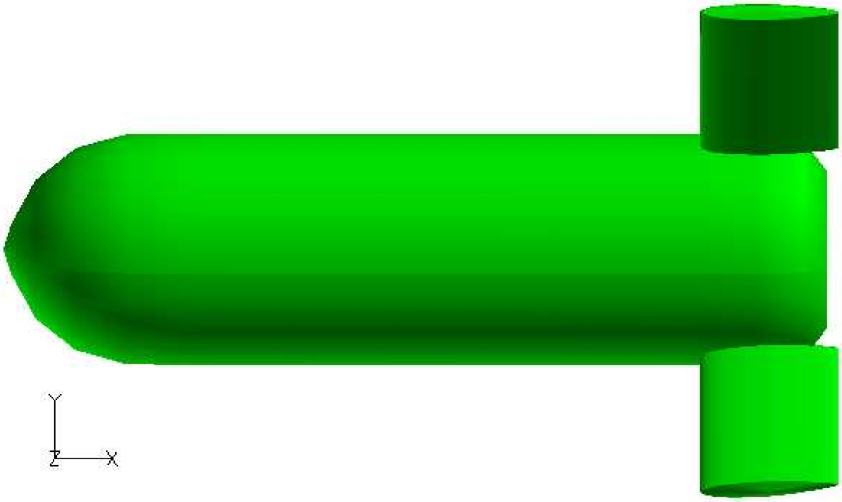

Figure 15: Generic Store Body Computational Model

Table 7: Generic Store Grid Dimensions

Grid	**Size**
Generic Store Body Superblock	53,000 cells
Generic Fin Superblocks	4 x 109,434 cells
Cartesian Interface Mesh	375,000 cells
Small Background Mesh	456,950 cells

read in four times to produce four fins around the store, each with its own force and dynamic specification. These five superblocks are assembled into a single super dynamic group, which is the store in its entirety.

These grids were obtained from the AF Seek Eagle Office and used in their unmodified form. The body surfaces are displayed in Figure 16 with a grid cutplane through the XZ plane to show the overlapping grids. The same small background mesh seen in Figure 13 is used as the background mesh in this problem. A different interface mesh is used, which was generated using Gridgen®. This interface mesh surrounds the generic store grids with more than enough room for grid interpolation

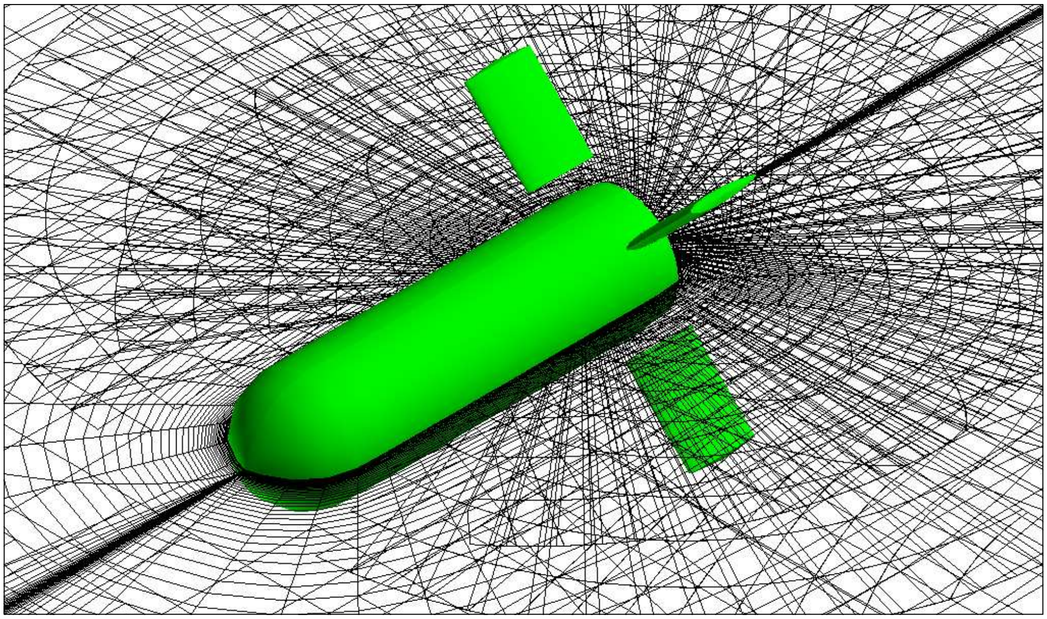

Figure 16: Grid Cutplane around Generic Store Body with Fins in "X" configuration

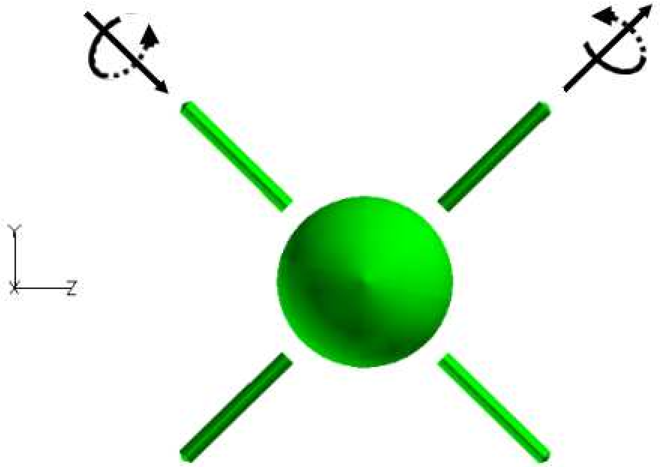

Figure 17: Positive Rotation Direction of Upper Fins

to the background mesh. The interface mesh was designed with fine cells to minimize grid interpolation errors. The dimensions of these grids are seen in Table 7; the system contains 22 grids and just over 1.3 million cells.

3.4.2 Generic Store Testing. The static solution around the generic store body is first obtained with the same solver conditions and time step ramping schedule used in the MK-84 model. The solution is run for 350 iterations to determine sufficient convergence of the flow field. The steadiness of the force and moment coefficients is used as the convergence criterion.

The same comparison approach will be used to verify the accuracy of the addition of moving components to a pinned store body. A single test case will be used with the initial conditions of Mach 0.6 at a standard altitude of 20,000 feet. The dynamic motion of the body will be controlled through three stages of motion. A simple pitch up motion will be prescribed on the store by rotating the fins in the direction shown in Figure 17 over the first 0.3 physical seconds. Over the next 0.3 seconds, the fins will be rotated back down to their initial position. For the remainder of the solution time, the fins will remain fixed with respect to the store body. This cycle is shown in Figure 18. Only the top two fins rotate in response to this cycle. The bottom two

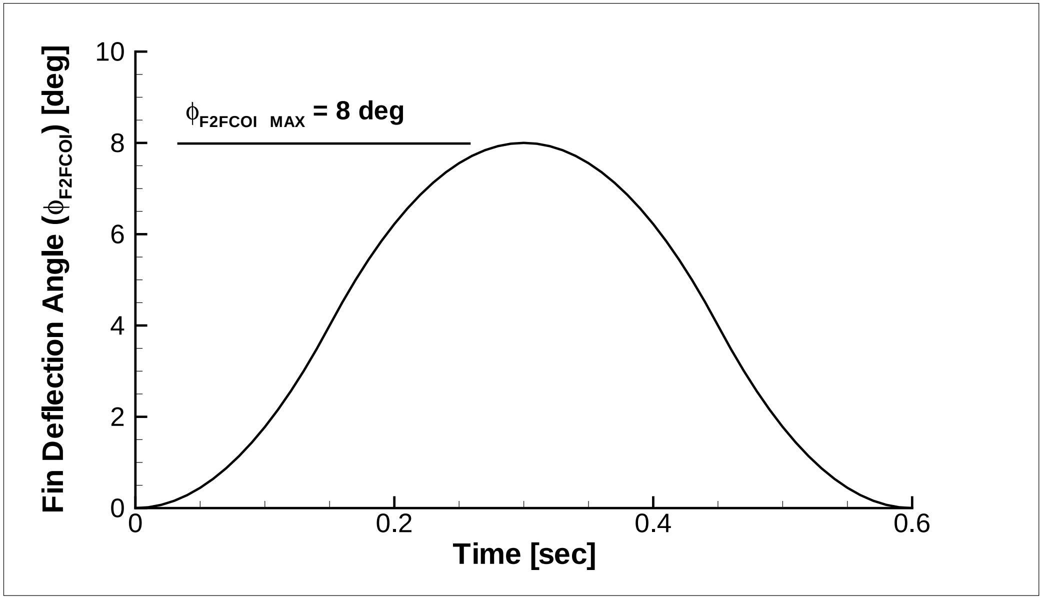

Figure 18: Prescribed Fin Deflection vs. Time Profile for Upper Fins

fins are held fixed throughout the simulation by increasing the motion start time far above the expected solution time. The details of the prescribed motion of each fin are shown in the dynamic specifications in Appendix B.4.

The goal of this testing is to simply confirm the motion of the pitching store between the two solution methods. The store's response to the applied force from the fins should be similar between the translating and pinned cases. The same numerical data output by Beggar used in the MK-84 test cases will be used here to verify the similarity of these solutions.

3.5 Beggar Inputs

Beggar inputs are split into several different files. The main input file contains such things as the Mach number, (6+)DOF inputs, flow solver options, and grid inputs. Other inputs such as the force specification and dynamic specification may be separated into other files that are referenced from the input deck, or written in the input deck itself. The force specification outputs force and moment coefficients to a data file at a user-specified frequency. The user inputs reference values for the body and selects regions on which the forces and moments are output.

The dynamic specification controls the dynamic and super dynamic groups. This specification contains the grids that are to move in response to the forces as well as the inertial properties of those bodies. In the multi-body problem, the dynamic specification of the main body must be followed by the dynamic specification of each moving component. These component specifications allow the user to input the properties and prescribed motions of the components separately.

The MK-84 AIR input file is shown in Appendix B, along with the file containing the boundary conditions for the MK-84 AIR body. The generic store input and boundary conditions files are similar. The force and dynamic specifications of the generic store problem are shown in Appendix B.4.

IV. Results & Discussion

Results from the three computational models are presented next. The first model, the supersonic compression ramp, is used in three test cases to verify the minor code modifications of background mesh motion and total flow visualization. The second model, the MK-84 AIR, is used in two test cases to compare the pinned solution to the reference solution in which the store translates. These two test cases are also used to simulate the MK-84 AIR in extended free flight. The final model, the generic store body with multiple moving components, is used to test the final modifications of the code in a single comparison test and an extended free flight simulation test.

4.1 Supersonic Compression Ramp

The ramp grid is used as a single mesh problem to test the ability of the modified code to apply mesh motion to the background mesh. In this problem, the ramp grid and the background mesh are one and the same, so any application of mesh motion to the background mesh will be easily viewable on the ramp. The flow visualization corrections, which output the total flow solution, are also verified in this model.

4.1.1 Ramp Convergence History. The L_2 convergence norm was analyzed to confirm the convergence of the flow field over the compression ramp. A reduction of four orders of magnitude was desired for full convergence. As seen in Figure 19, the convergence norm descended four to five orders of magnitude for each test case within 1000 iterations. Accordingly, the test cases are considered to be fully converged. This rapid convergence is typical for supersonic flow over a compression ramp.

4.1.2 Ramp Results. The three test cases in Table 3 are used to evaluate success of application of mesh motion to the background mesh. The first case consists of pure flow motion and is used as a reference case to check the Beggar solution. With this Mach 2.0 flow over the 15 ° ramp, the exact solution is easily obtained from the oblique shock relations [4]. The resulting flow field should have a shock present at

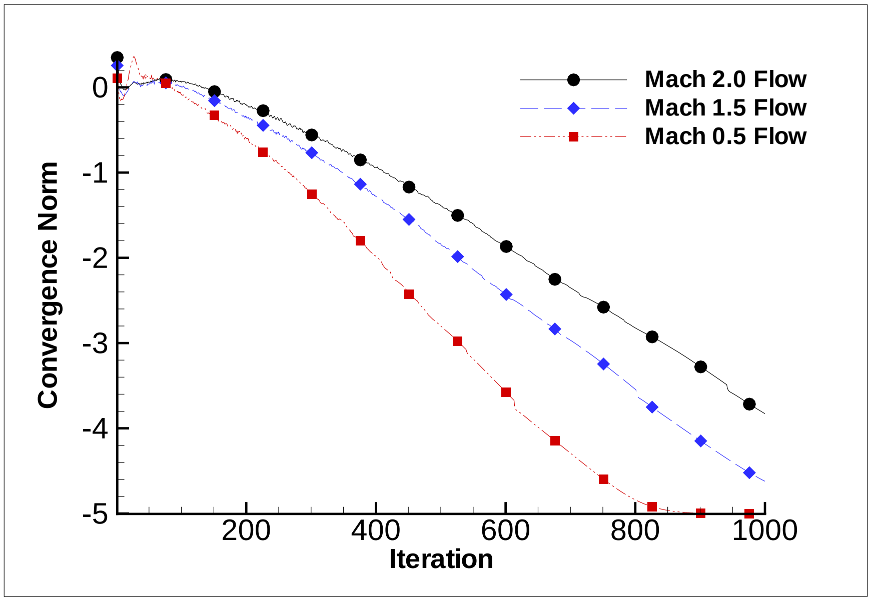

Figure 19: Convergence History of the Compression Ramp

a 45° angle, with a Mach number of 1.45 behind the shock. Beggar produces this solution, seen in Figure 20. There is a noticeable reflection behind the shock due to the imperfect far field boundary conditions on the upper surface.

The next two cases consist of lower levels of flow motion and some amount of mesh motion. In these cases, Beggar is programmed to apply a speed to the background mesh (the ramp) to bring the total flow to Mach 2.0. In the second case, the input Mach number is 1.5 and the mesh Mach number is programmed to

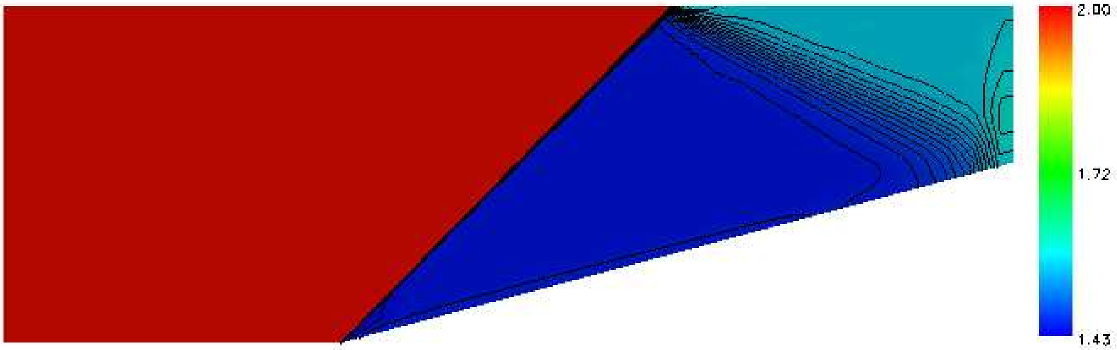

Figure 20: Ramp Case 1: Mach 2.0 Flow, No Mesh Motion

be 0.5; and in the third case, the input Mach is 0.5 and the mesh Mach is 1.5. Both of these cases result in similar flow solutions as the pure flow case. To demonstrate this, a plot of the Mach number in the x direction across the ramp for all cases is shown in Figure 21. It can be seen that with each combination of mesh and flow motion, a shock wave of similar strength is produced in a similar location over the ramp. The exact location of the shock and Mach number behind the shock (M_2) does vary slightly between solutions. This may be due to the closeness and small size of the grid cells around the shock location, causing that location to vary slightly and produce a slightly different M_2. Plots showing the contours of Mach number across the compression ramp for the second two cases are shown in Appendix A.1.

This confirms that this method of applying the mesh motion to the background mesh is effective and accurate. In addition, the flow visualization corrections are also shown to be accurate through this test case.

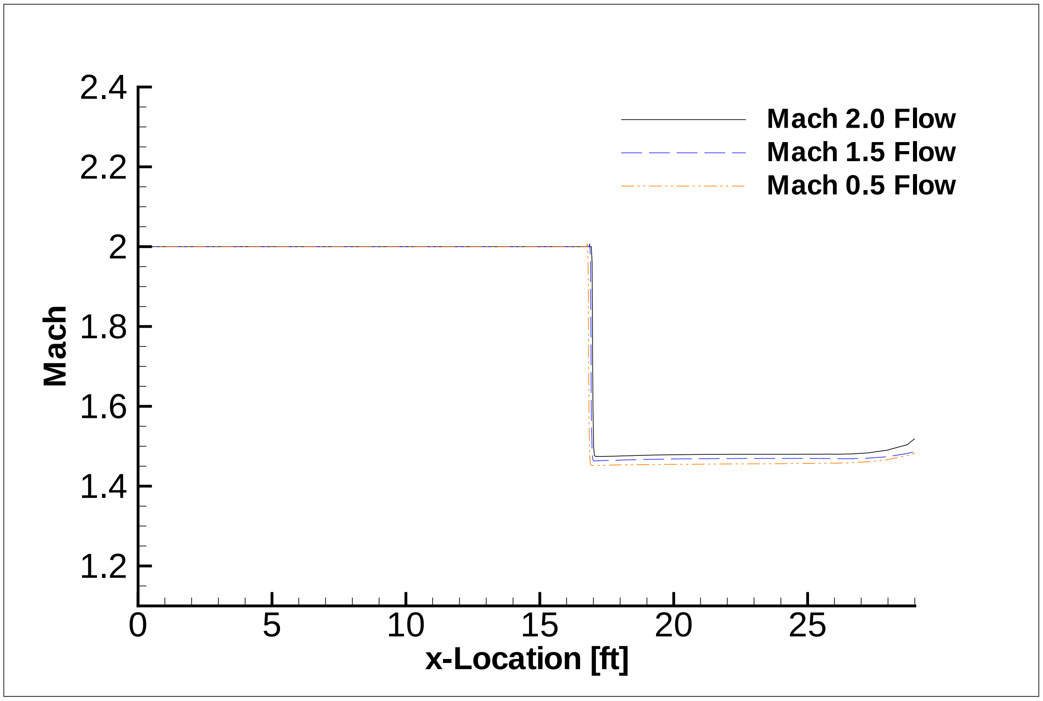

Figure 21: Comparison of Mach number across the ramp with varying amounts of mesh motion

Table 8: Standard Deviations of Forces and Moments in the Final 200 Iterations of the MK-84 AIR Static Solution

	Fx	**Fy**	**Fz**	**Mx**	**My**	**Mz**
Std. Deviation:	6.77E-4	8.18E-4	2.17E-4	1.35E-5	6.49E-4	2.57E-3

4.2 MK-84 AIR Results

4.2.1 Convergence History. Following the static solution approach outlined in Section 3.3.3, the forces and moments on the store in the small background mesh are observed to completely level out in 500 time steps (Figures A.3- A.4). The standard deviations of the forces and moments in the final 200 iterations are shown in Table 8. Because of these low standard deviations of the forces and moments, the solution is considered to be converged after 500 iterations. This same procedure is used to obtained the static solutions for the store in the tall background meshes, with similar results.

4.2.2 Comparison Test Results. All test cases are initialized from the appropriate converged static solution. The reference case refers to the old method whereby the store is translating through the tall background mesh in response to the forces acting on it. The modified method of free flight simulation where the store remains fixed in the background mesh is referred to as the pinned method. The translating method is run for the longest amount of time possible, remembering that this time is limited by the height of the background mesh. The pinned case uses the smaller background mesh and is manually restricted to the same amount of solution time.

4.2.2.1 Test Case 1. The first test case was run at an initial Mach of 0.6 at 20,000 feet for 2.37 seconds. The store was observed to fall approximately 90 feet vertically while accelerating to Mach 0.603. The impressive amount of spin stabilization in the MK-84 design is seen by the high roll rate of the store, which quickly accelerates to almost -8 radians per second after release because of the asymmetric fins.

The force and moment coefficients acting on the store are shown in Figures 22 and 23. These show reasonable agreement between solution methods. However, there is increasing disparity between the methods as time progresses. Despite these differences in the forces and moments, the velocities show excellent agreement between methods throughout the duration of the solution (Figures 24 and 25). The periodic nature of the pitch and yaw rates of the store is clearly seen in Figure 25, and the data matches well throughout these oscillations.

The G&C telemetry data is perhaps the most impressive result, and also one of the most important. In both trajectory and orientation, the pinned method very accurately matches the solution from the translating method. Figures 26 and 27 show these results over the duration of the solution. The high amount of spin stabilization is seen as the MK-84 rolls through almost two complete revolutions in the 2.37 seconds after release.

The difference between the forces and moments calculated by two methods grows as the solution progresses in time. This trend is also seen in the translational velocity data and the trajectory. In the translating case, the store grids and all grids associated with the dynamic group must interpolate their values from the background mesh as they move. As this happens, discrete variations in the solution result which contribute to error in the solution.

As the store moves, the solid body uncovers holes. These holes are areas where a numerical solution was previously not possible because the cells were inside the solid body. As a hole is gradually uncovered over several time steps, it is treated as a fringe point. The values for these fringe points are interpolated from the surrounding cells and thus contain an interpolated value before they completely rejoin the computational domain. The translating case must interpolate values much more as the store is displaced more than in the pinned case. This interpolation difference may contribute to differences in the solution.

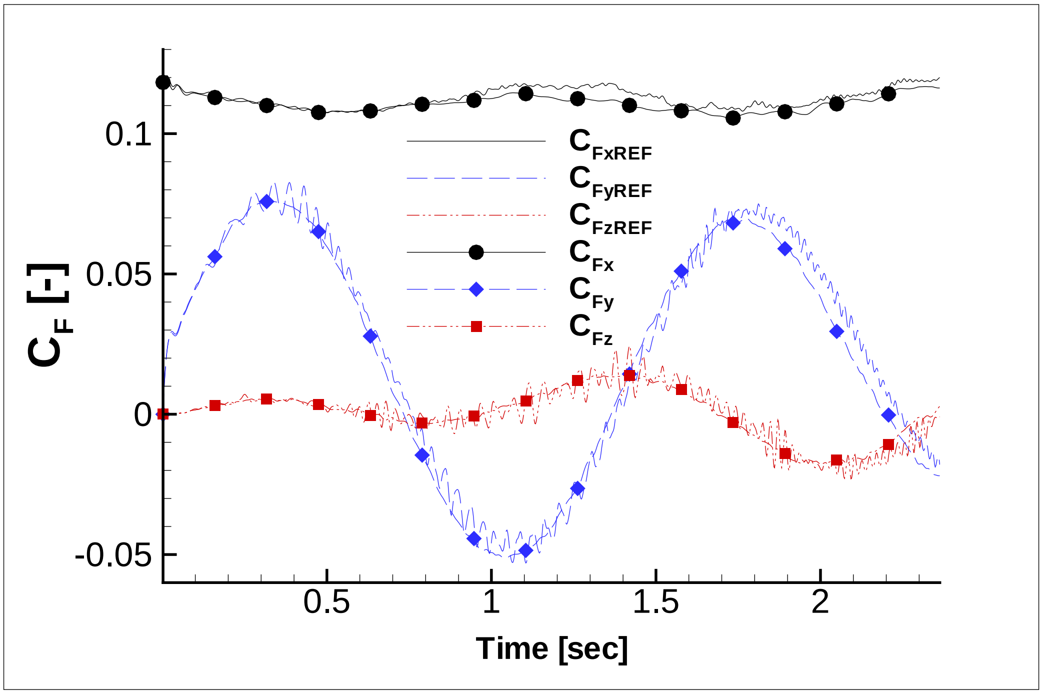

Figure 22: Force History on MK-84 AIR body at Mach 0.6 at 20,000 ft

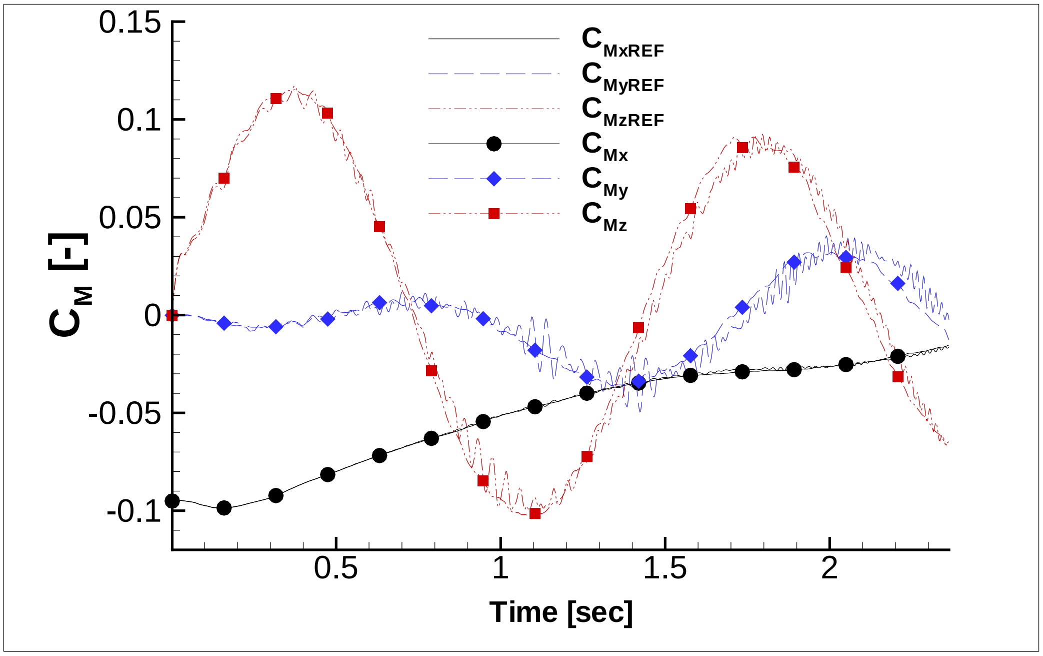

Figure 23: Moment History on MK-84 AIR body at Mach 0.6 at 20,000 ft

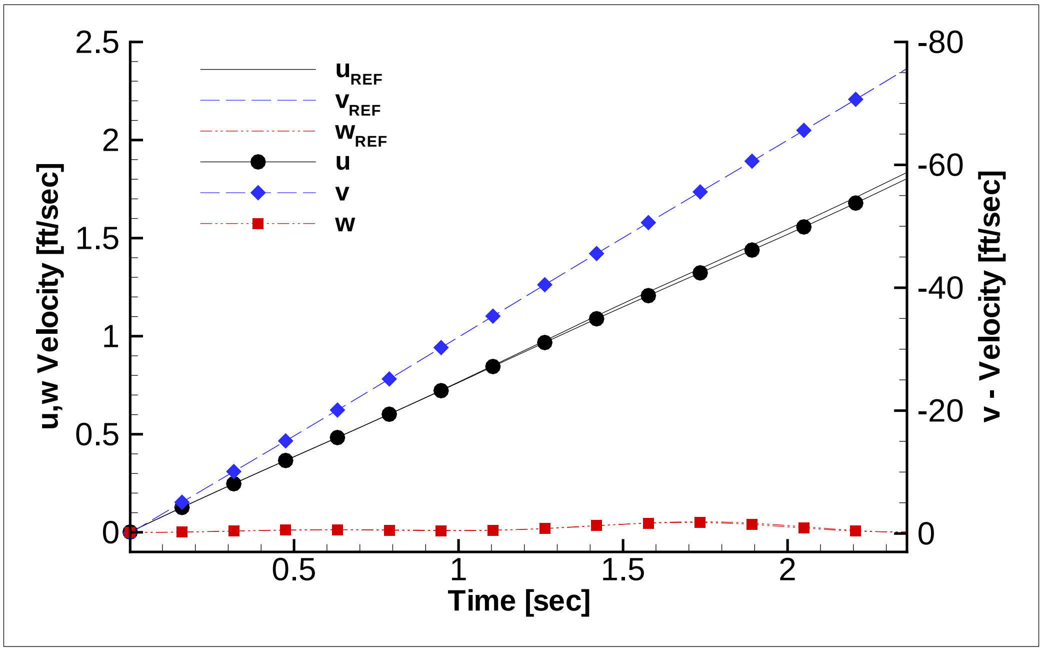

Figure 24: MK-84 AIR Velocity History at Mach 0.6 at 20,000 ft

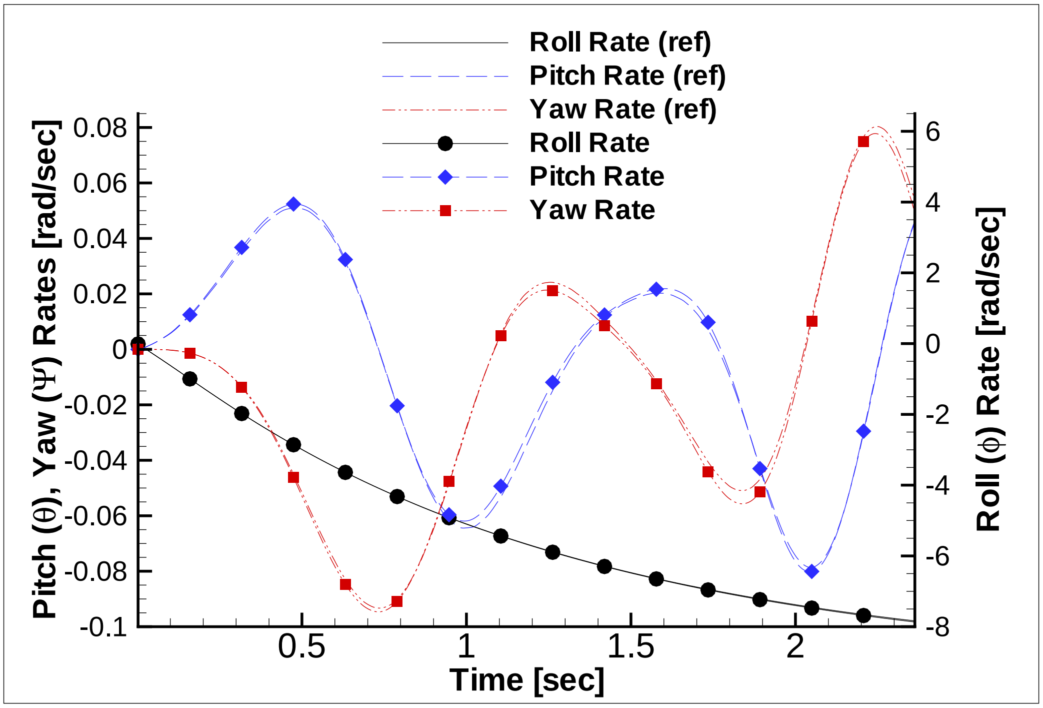

Figure 25: MK-84 AIR Angular Velocity History at Mach 0.6 at 20,000 ft

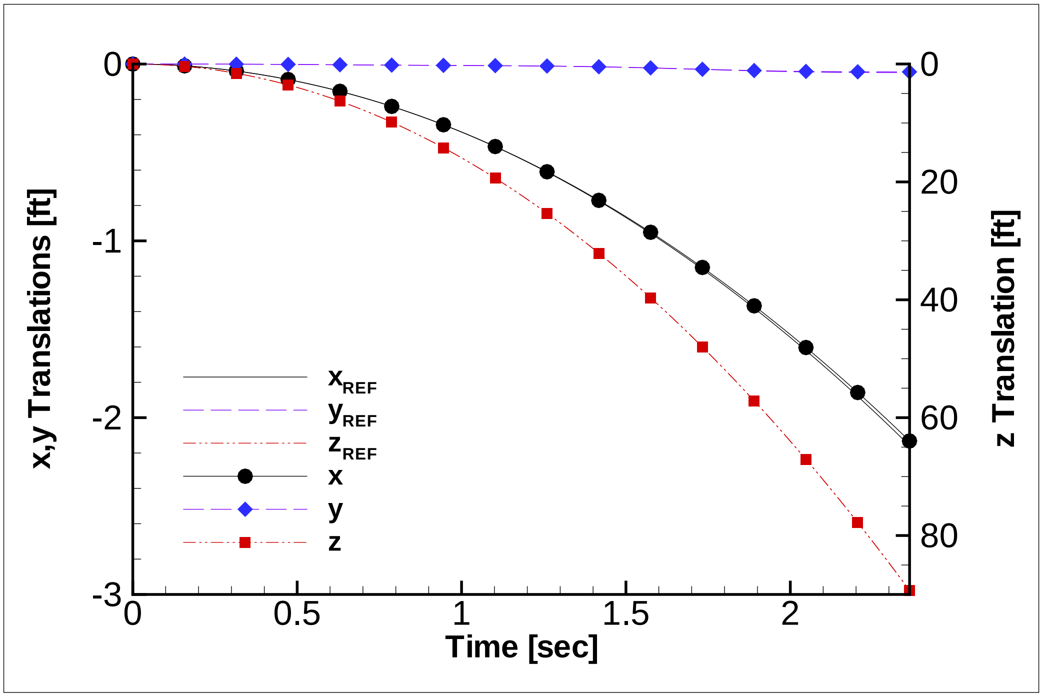

Figure 26: MK-84 AIR Trajectory at Mach 0.6 at 20,000 ft

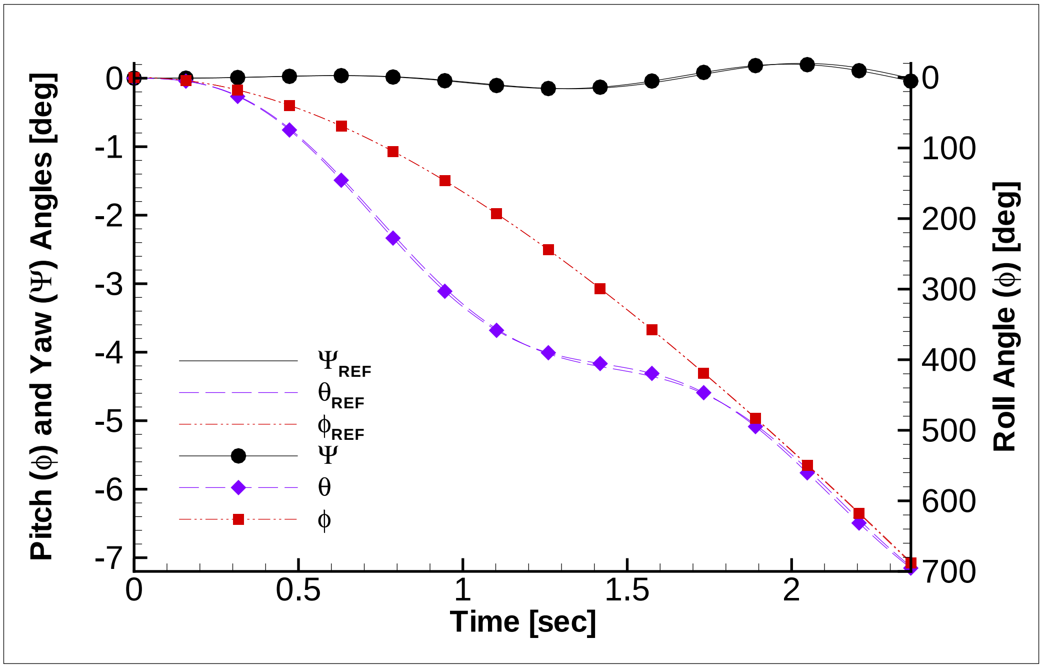

Figure 27: MK-84 AIR Orientation at Mach 0.6 at 20,000 ft

In addition, the store is initialized in the center of the background mesh, giving the maximum distance between the store and the boundaries. As the store translates away from this ideal position, it moves closer to the boundaries. Although care was taken to ensure adequate distance at all times, these boundary effects may also contribute slightly to error in the solution of the translating case.

These differences are especially seen in the forces and moments (Figure A.5 and A.6). The forces from the translating case are quite oscillatory, whereas the pinned case results in smoother forces following the similar trends. The same tendency is seen in the moments. Although the pinned case also moves and must interpolate new positions, these movements are in response to rotations only and thus require smaller interpolation jumps, causing less error.

Because the maximum difference between methods is at the end of the solution time, the differences at this point are analyzed. The percent differences are found by dividing the difference by the average value. The forces and moments for Case 1 are shown in Table 9. Most of these percent differences are very small, with the exception of the coefficients C_{Fz} and C_{My}. These percent differences are much larger because the average measured value lies close to zero. Though these percentages are large, the actual deviation between the values is still very small.

Similar small differences in the velocities are seen in Table 10, with the exception of the w component of velocity. The same reason applies: even though the actual difference is small, the percentage is large because this velocity component is so close to zero. All telemetry differences are also small (Table 11), with the exception of the yaw angle (ψ). Even though the pinned yaw angle is an order of magnitude different than the translating angles, both methods result in a near-zero yaw angle. Because the store is flying straight through the air, it makes sense that this yaw angle would be close to zero.

4.2.2.2 Test Case 2. Test case 2 (initially $M = 0.9$, 10,000 ft) was run for 2.27 seconds of solution time. The MK-84 behavior in this case is slightly

Table 9: MK-84 AIR Test Case 1: Forces and Moments after 2.36 seconds

	Force Coefficient [-]			Moment Coefficient [-]		
	x	y	z	x	y	z
Ref Case:	0.1200	-0.0180	-8.90E-04	-0.0163	-0.0024	-0.0651
Pinned Case:	0.1163	-0.0219	2.90E-03	-0.0153	-0.0128	-0.0676
Difference (%):	3.16	19.78	200.00	6.52	138.12	3.65

Table 10: MK-84 AIR Test Case 1: Velocities after 2.36 seconds

	Velocity [ft/s]			Angular Velocity [rad/s]		
	u	v	w	$\dot{\phi}$	$\dot{\theta}$	$\dot{\psi}$
Ref Case:	1.8347	-75.6352	5.85E-04	-7.8349	0.0463	0.0537
Pinned Case:	1.8029	-75.6842	2.55E-03	-7.8669	0.0456	0.0493
Difference (%):	1.75	0.06	125.38	8.61	1.42	0.41

Table 11: MK-84 AIR Test Case 1: Trajectory and Orientation after 2.36 seconds

	Translation [ft]			Orientation [degrees]		
	x	y	z	ψ	θ	ϕ
Ref Case:	-2.1584	-0.0481	89.2929	-0.00403	-7.1293	686.712
Pinned Case:	-2.1318	-0.0443	89.3350	-0.04256	-7.1515	688.159
Difference:	0.0266	0.0038	-0.0421	-0.0385	-0.0223	-1.447
Difference (%):	1.24	8.15	0.05	165.41	0.31	0.21

different but realistic considering the different initial conditions. The store falls a similar vertical distance, but pitches less and rolls much more. In fact, the roll rate of the store is over double the first test case. This is expected because the magnitude of the forces acting on the fin is larger due to the greater density and velocity in this case.

Plots showing the time histories of the pertinent data from this case are shown in Appendix A.3 and the differences at the end of the solution time are tabulated in Tables 12 to 14. The agreement between the force and moment coefficients in this case is much better; the highest percent difference seen is 44% in the C_{Mz} component (Table 12). With the exception of the spike in percent differences because of the near-zero values of yaw and pitch velocities, the velocities of the store show much better agreement between solutions than the forces and moments do (Table 13). Very low differences are seen in the trajectory and orientation between the two methods. The percent difference between the yaw angle is very large as both values are near zero.

Overall, the two test cases show good agreement between the translating and pinned solutions. In both cases, the forces and moments seem to be the most different. However, this difference does not have a large effect on the velocities and telemetry, which are very similar between solution methods.

Table 15 shows the total amount of wall clock time needed for the 1200 time steps used in the both solution methods of these cases. The benefit of using the

Table 12: MK-84 AIR Test Case 2: Forces and Moments after 2.27 seconds

	Force Coefficient [-]			Moment Coefficient [-]		
	x	y	z	x	y	z
Ref Case:	0.1149	0.0258	-1.12E-02	-0.0009	0.0116	0.0213
Pinned Case:	0.1133	0.0203	-9.31E-03	-0.0008	0.0074	0.0189
Difference (%):	1.37	23.78	18.78	8.48	44.00	12.03

Table 13: MK-84 Case 2: AIR Test Velocities after 2.27 seconds

	Velocity [ft/s]			Angular Velocity [rad/s]		
	u	v	w	$\dot{\phi}$	$\dot{\theta}$	$\dot{\psi}$
Ref Case:	6.0640	-72.5411	-3.81E-02	-0.0144	-0.0439	-15.2003
Pinned Case:	6.0037	-72.7319	-4.73E-02	-0.0439	-0.0139	-15.2012
Difference (%):	1.00	0.26	21.71	101.17	103.70	0.01

Table 14: MK-84 AIR Test Case 2: Trajectory and Orientation after 2.27 seconds

	Translation [ft]			Orientation [degrees]		
	x	y	z	ψ	θ	ϕ
Ref Case:	-6.8403	-0.0047	82.3965	0.07656	-4.1965	1586.8100
Pinned Case:	-6.7927	0.0176	82.7352	0.06802	-4.1970	1588.1415
Difference (%):	0.70	200.00	0.41	11.81	0.01	0.08

modified code with the 24% smaller background mesh is seen here as a 17% reduction in solution time. This reduction is the same for both test cases.

4.2.3 Free Flight Simulation Results. Once the modified code was verified, extended simulations of the MK-84 AIR in free flight were conducted. The first test case simulated was the MK-84 AIR with an initial Mach of 0.6 at sea level standard conditions, depicted in Figure 28. This simulation lasted for 135 seconds of solution

Table 15: Wall Clock Times for MK-84 AIR Test Cases

	Case 1	**Case 2**
Translating Method	5.9 hrs	5.9 hrs
Pinned Method	4.9 hrs	4.9 hrs
Gain	17%	17%

time, taking 74,000 iterations. This required 324 hours (13.5 days) of wall clock time using eight processors.

The store fell 167,000 feet in this time, reaching a final pitch angle of 86.2 degrees nose down. The store accelerates to Mach 1.47, as shown in Figure 29, but does not appear to reach a conclusive terminal velocity speed. This Mach number is far above the experimental terminal velocity of Mach 1.03. The difference is attributed to the lack of viscous effects in this simulation, which removes the parasitic drag component.

The MK-84 AIR remains stable throughout the simulation. The pitch and yaw oscillations immediately following the release are the largest, and become damped out as the solution progresses. The supersonic regime is entered in 31 seconds as seen in Figure 30. This is accompanied by the expected wave drag rise in both the x and y directions (Figure 31). After the store goes supersonic, the y component of drag continues to increase as the store accelerates and pitches nose down. As the store continues to pitch nose down, the y component of drag will continue to become greater until it is balanced by the gravitational force at the terminal velocity speed.

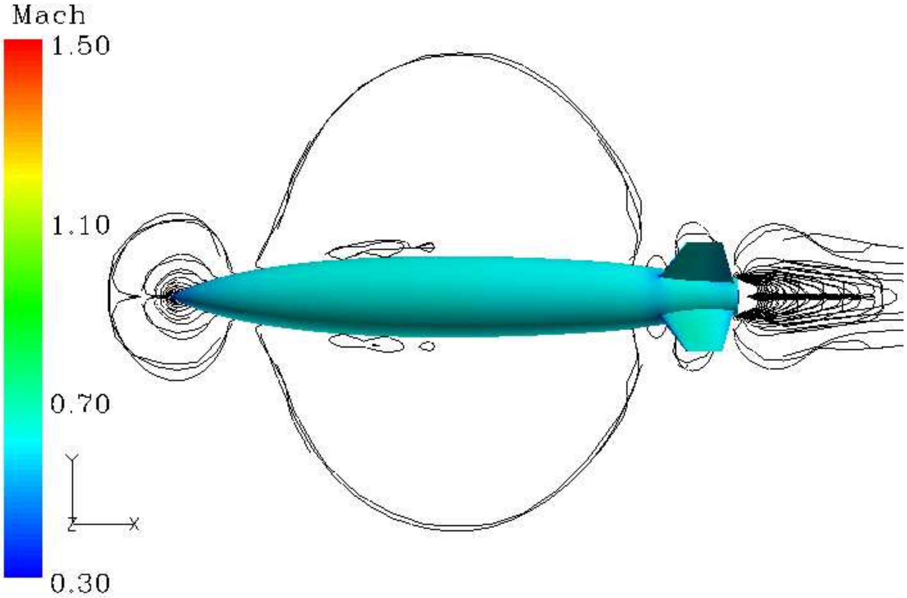

Figure 28: Contours of Mach showing the MK-84 AIR in its pre-release position at Mach 0.6 at Sea Level

The moments show the wave drag even more clearly than the forces do. As seen in Figure 32, there is a sharp rise in the y and z moment coefficients as the store passes the sound barrier. This may be due to a non-zero angle of attack as the store passes the sound barrier. Any shock wave forming on the store with a non-zero angle of attack would produce asymmetric wave drag, leading to increased moments. This rise is quickly damped out in less than 10 seconds and the moments continue to damp slightly for the rest of the solution.

This effect can also be seen in the history of the angular velocities, shown in Figure 33. The pitch and yaw oscillations begin to damp out as the simulation begins, until the supersonic point is reached. Then there is a small spike in the pitch and yaw rates, which quickly damps out. These rates continue to damp well into the solution, showing the stability of the MK-84 AIR.

The history of the position and orientation of the store are shown for completeness. It is interesting to note that the y translations (in the G&C coordinate frame),

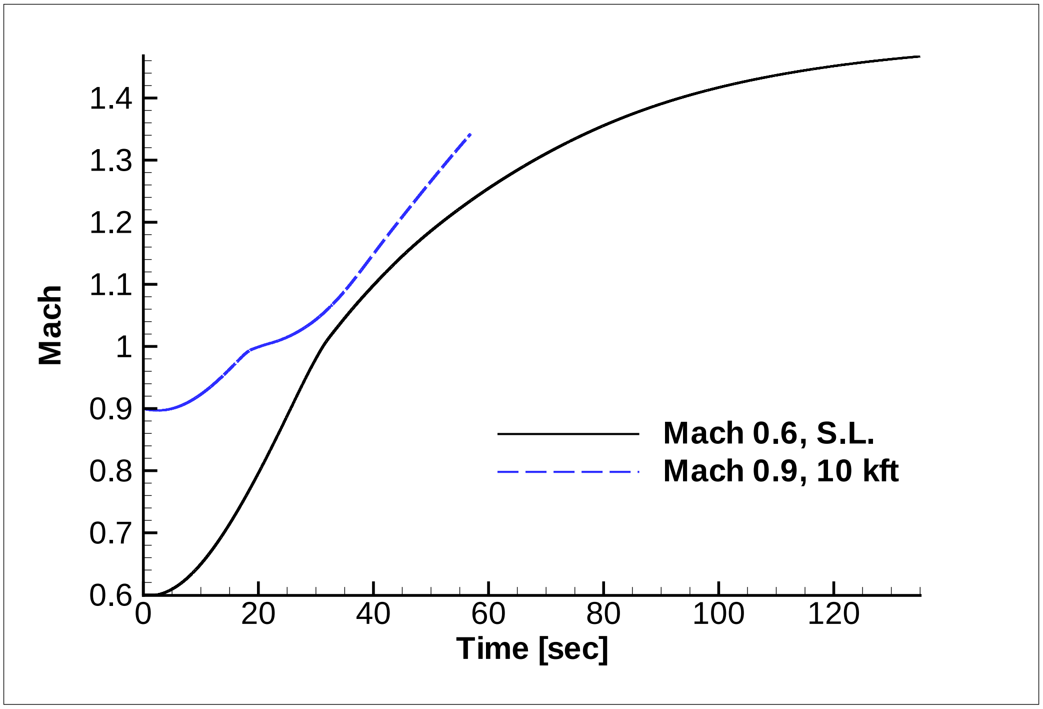

Figure 29: Mach History of Extended MK-84 AIR Simulations

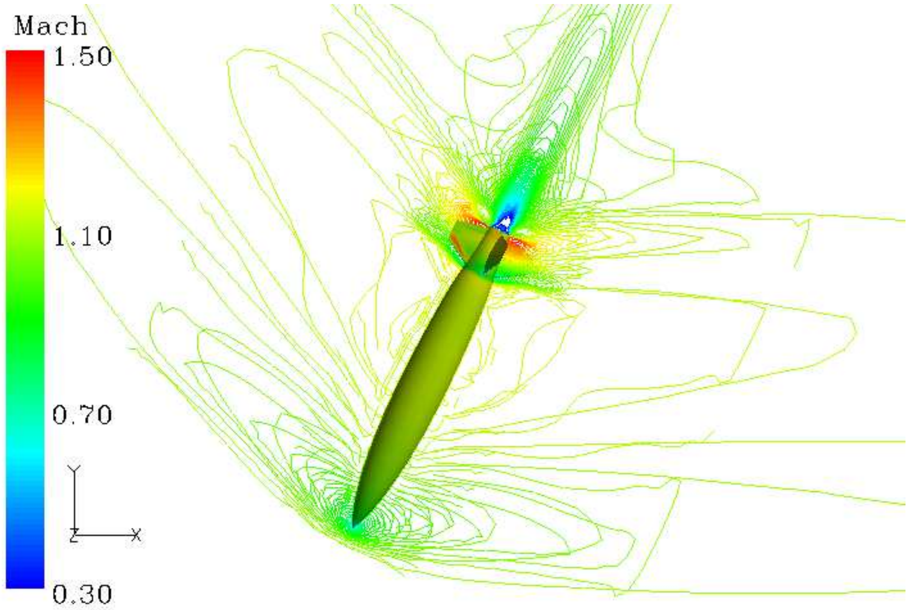

Figure 30: Contours of Mach showing the MK-84 AIR just after entering the supersonic flight regime

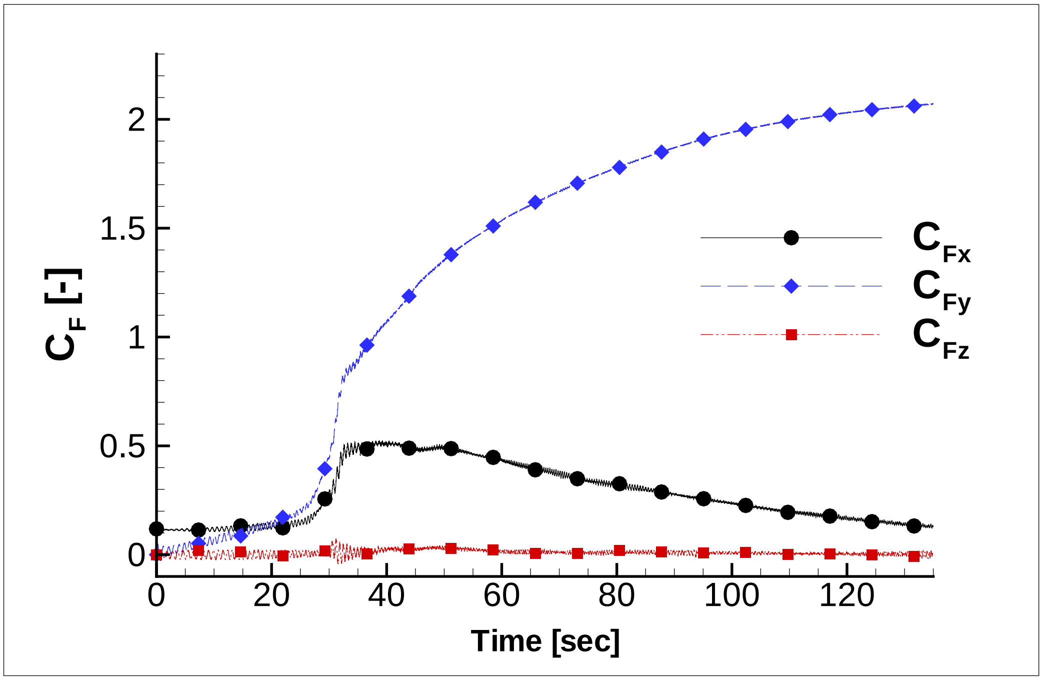

Figure 31: Extended Force History on MK-84 AIR body at Mach 0.6 at Sea Level

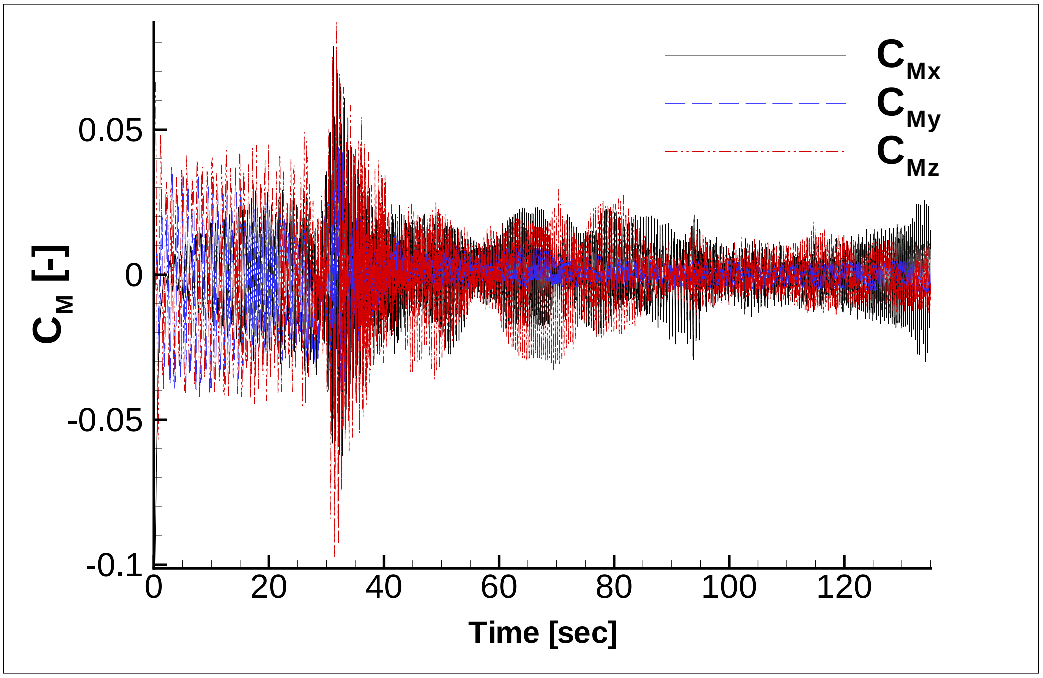

Figure 32: Extended Moment History on MK-84 AIR body at Mach 0.6 at Sea Level

reach a maximum of about 900 feet by the end of the solution (Figure 34). This shows that the MK-84 AIR did not fly perfectly along its original heading, but deviated slightly over the course of the simulation. The orientation history shows that the store is approaching a 90° nose down orientation, with the maximum reached here being 86.2° nose down (Figure 35). The non-zero yaw component is also clearly seen here. The history of the total roll angle is not shown because of the extremely large values reached.

Contours of Mach number showing the flow field around the MK-84 in its final position at the end of this simulation are shown in Figure 36.

The second test case, Mach 0.9 at 10,000 feet, is run for 56.8 seconds of solution time in 30,000 iterations, with much more interesting results. The store falls 43,069 feet with a final pitch angle of -67.7 degrees. The spin stabilization of the store follows a profile similar to the first case, as seen in the history of the roll rate. The sound barrier is reached in only 20.2 seconds due to the higher initial Mach. The store

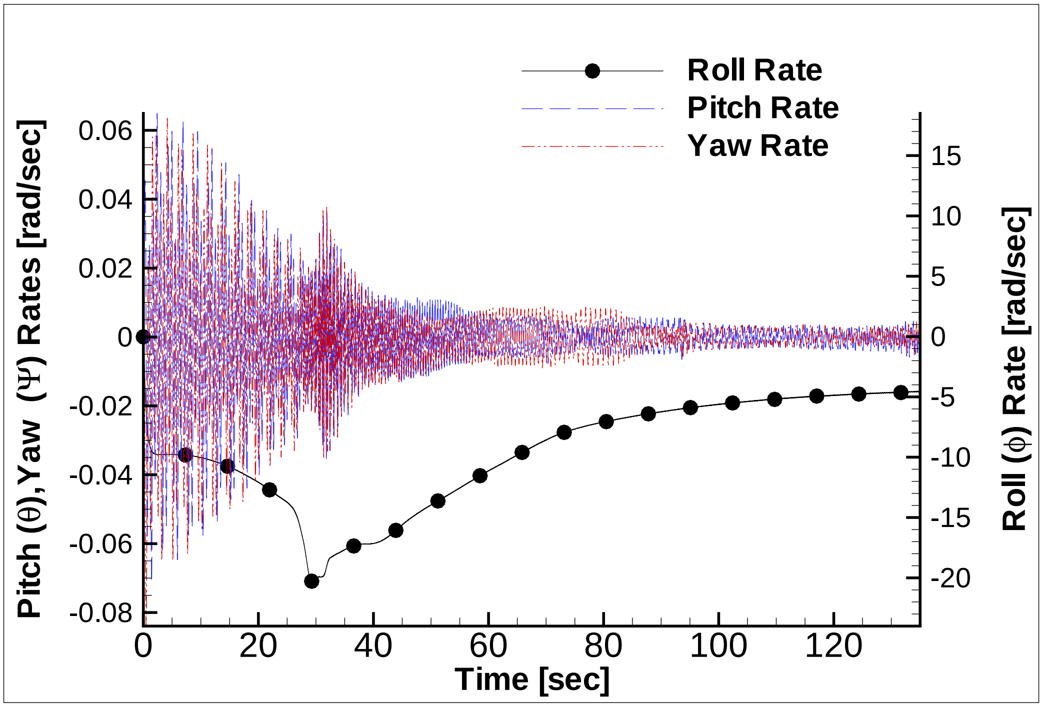

Figure 33: Extended Angular Velocity History of MK-84 AIR at Mach 0.6 at Sea Level

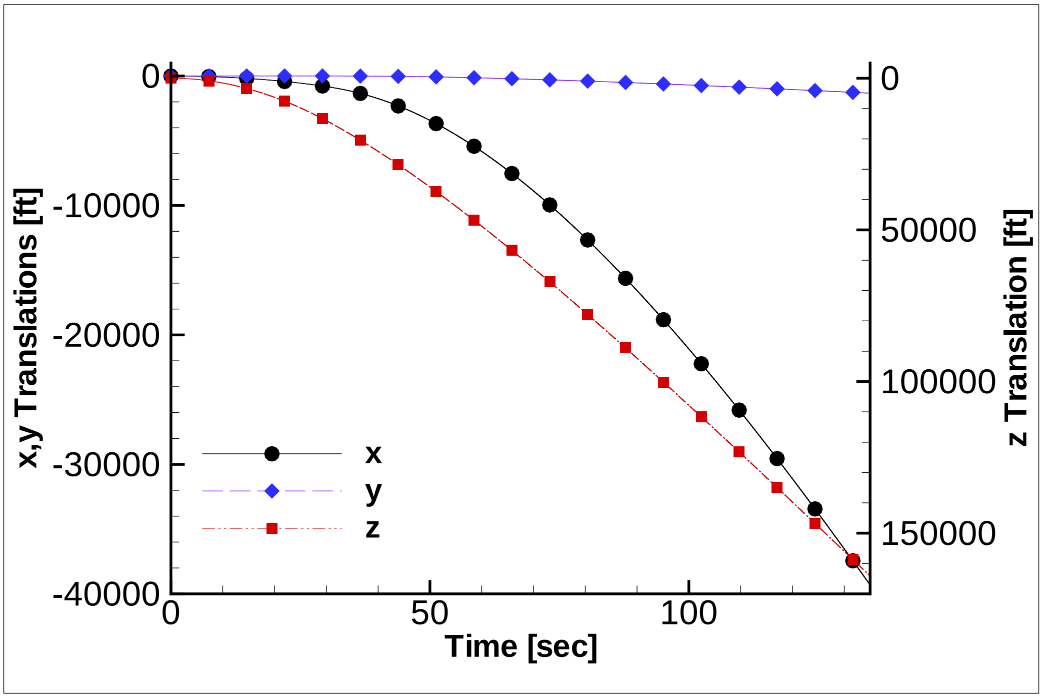

Figure 34: Extended Trajectory of MK-84 AIR at Mach 0.6 at Sea Level

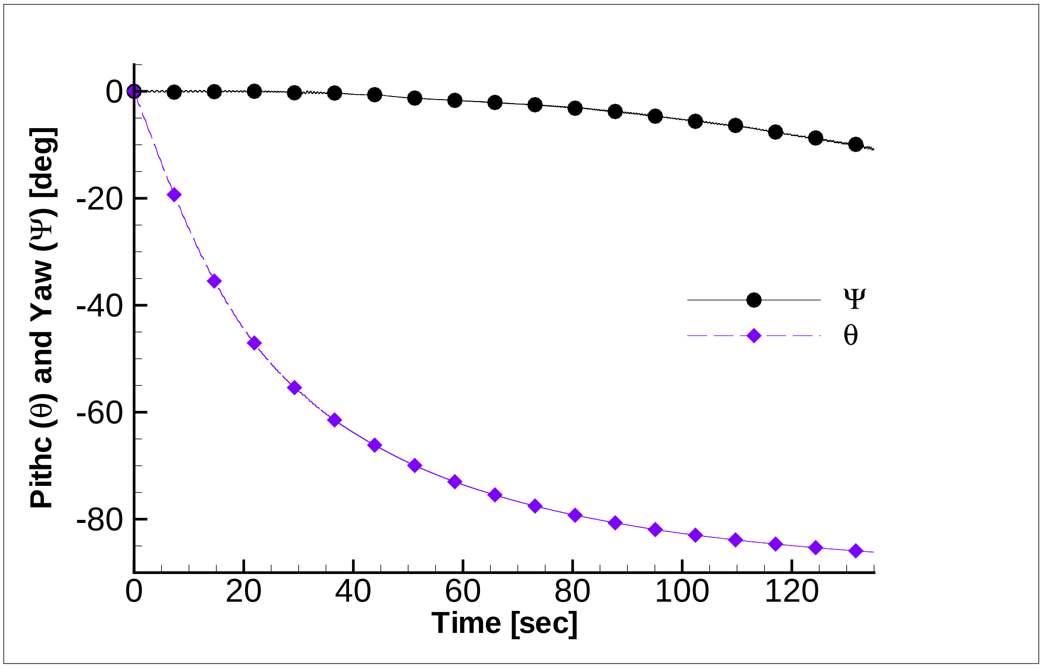

Figure 35: Extended Orientation History of MK-84 AIR at Mach 0.6 at Sea Level

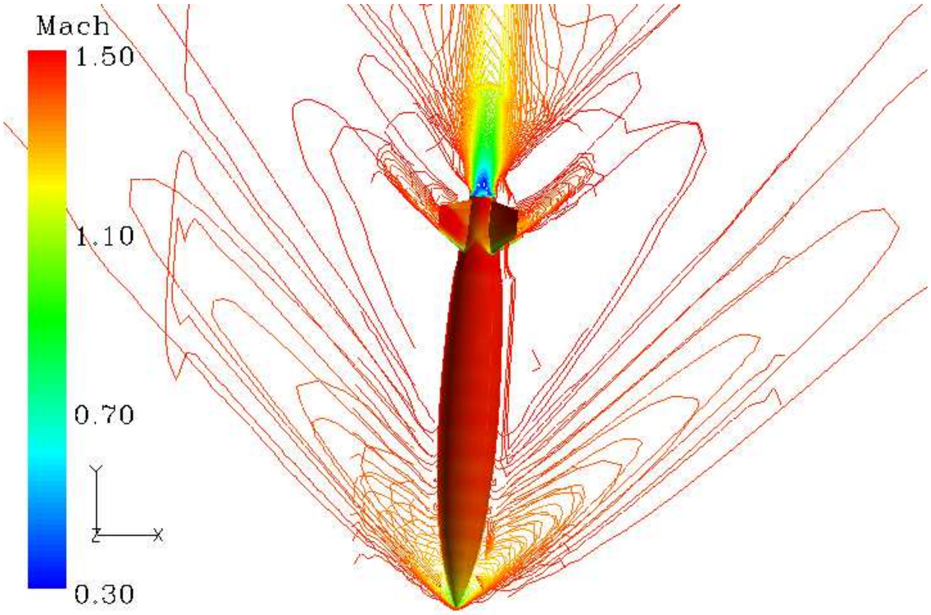

Figure 36: Contours of Mach around the MK-84 AIR after 135 seconds of free flight

accelerates to Mach 1.34, with no signs of leveling off at a terminal Mach number as shown in Figure 29.

The solution in the transonic and supersonic region is especially interesting. The forces and moments on the store body are shown in Figures 37 and 38. The amplitude of the force and moment oscillations begins to increase slowly in the transonic region. The flow in this regime is already reaching supersonic speeds as it passes over the curvature of the body. As the store approaches the sonic point, the yaw and pitch oscillations increase dramatically. This happens at approximately 18 seconds into the solution. The store appears to be departing from stable flight; however, as the solution continues the stability of the MK-84 is observed. The forces and moments begin to damp out, until almost 30 seconds later when they return to values similar to those seen before the instability occurs.

Similar behavior is seen in the pitch and yaw rates of the store (Figure 39). These rates also encounter an instability near the sonic point, which damps out as the simulation continues. The pitch amplitude during this period is +/- 10 degrees

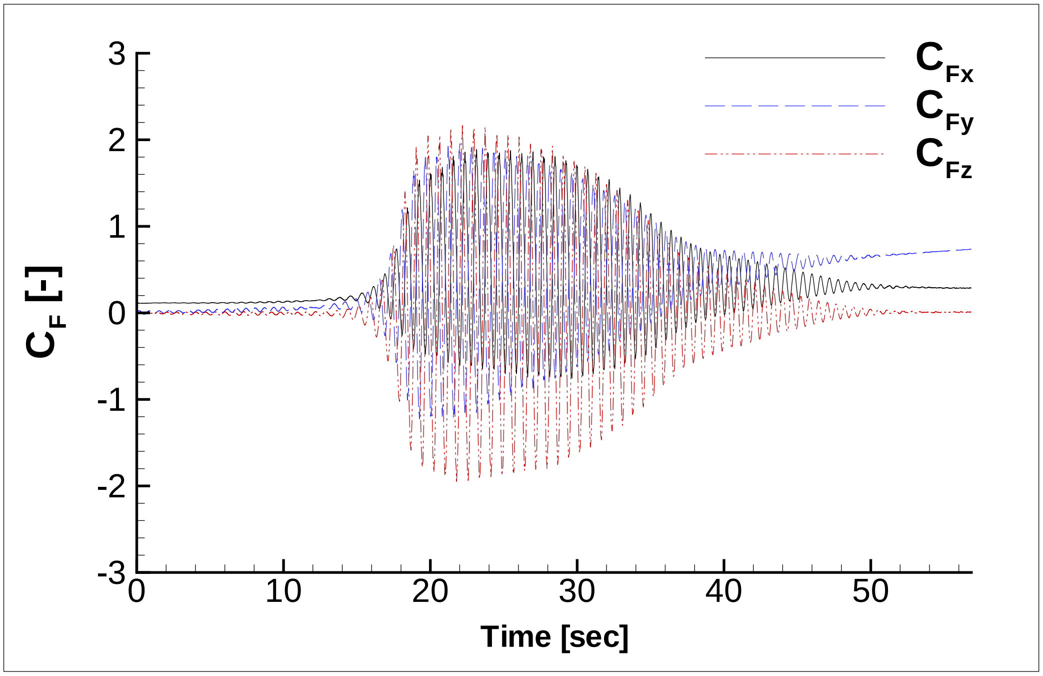

Figure 37: Extended Force History on MK-84 AIR body at Mach 0.9 at 10,000 ft

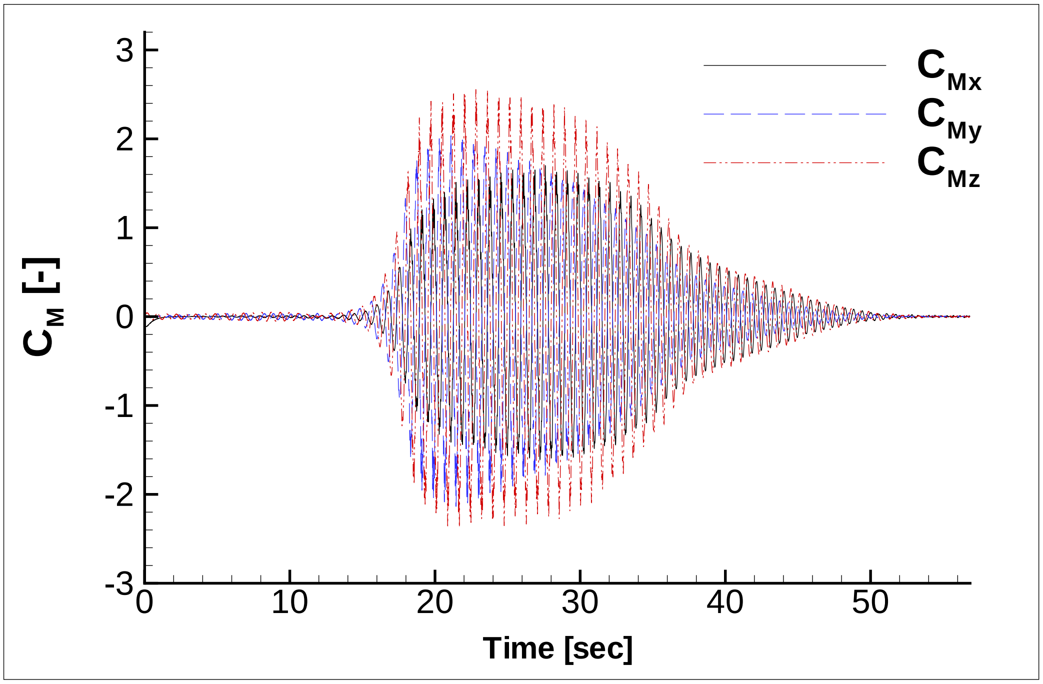

Figure 38: Extended Moment History on MK-84 AIR body at Mach 0.9 at 10,000 ft

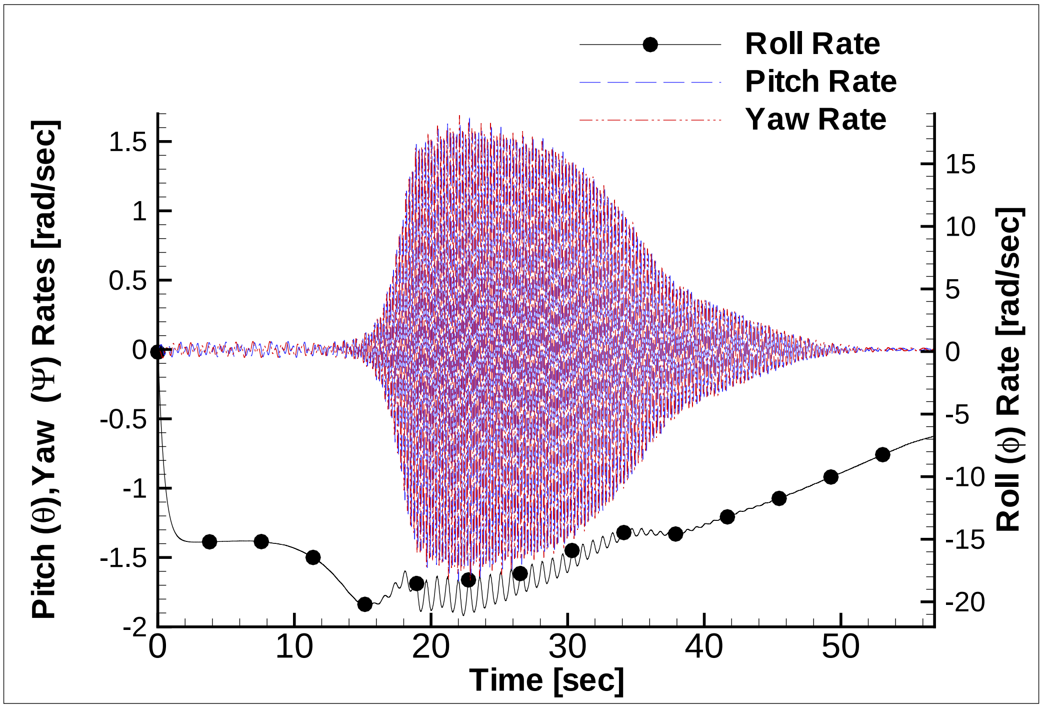

Figure 39: Extended Velocity History of MK-84 AIR at Mach 0.9 at 10,000 ft

and the yaw amplitudes are +/- 15 degrees around zero (Figure 41). Once damped, the solution continues in a stable fashion until it ends at 56 seconds.

The increased forces during this time slows the acceleration of the store, which can be seen from the plot of Mach number vs. Time in Figure 29. It is not until the large oscillations damp out that the acceleration returns to the value initially seen in the solution. This may be one reason this test case does not appear to approach a terminal velocity similar to the first test case. The trajectory of this case is depicted in Figure 40, and the orientation throughout the simulation is shown in Figure 41.

The exact cause of these unexpected oscillations is unknown. One contributing factor may be the formation of asymmetric wave drag as the store passes the sonic point. However, this wave drag did have such an affect on the first test case. An analysis of the pressures around the store during this event revealed little. A small imbalance in the distribution of pressures around the fins was observed slightly before this event occurred; however, what causes this imbalance is unknown. Nevertheless,

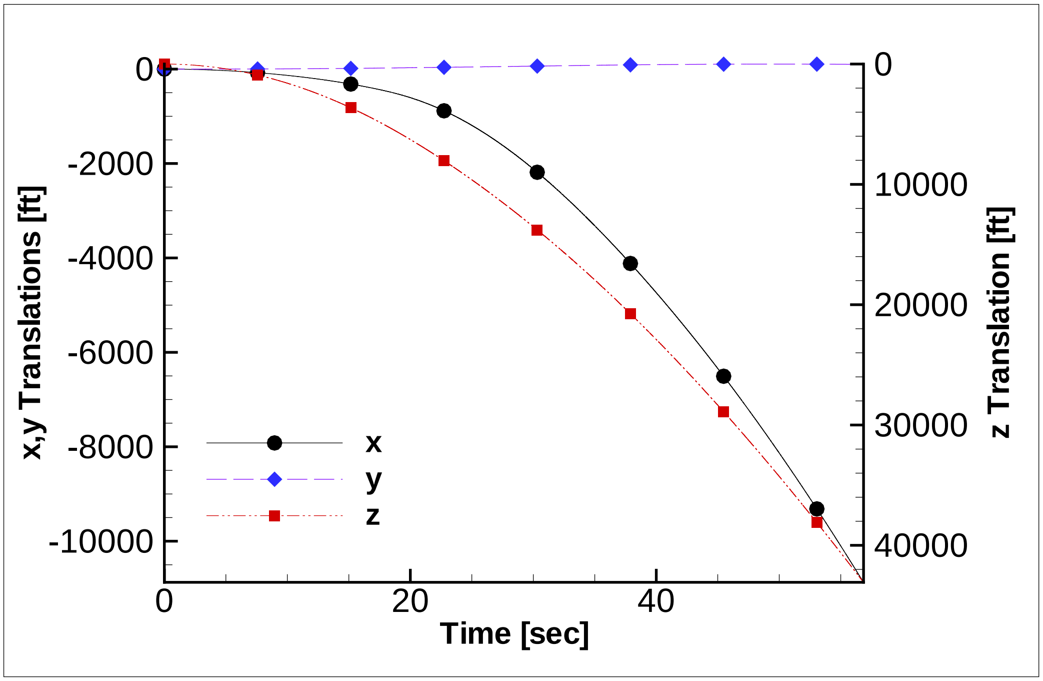

Figure 40: Extended Trajectory of MK-84 AIR at Mach 0.9 at 10,000 ft

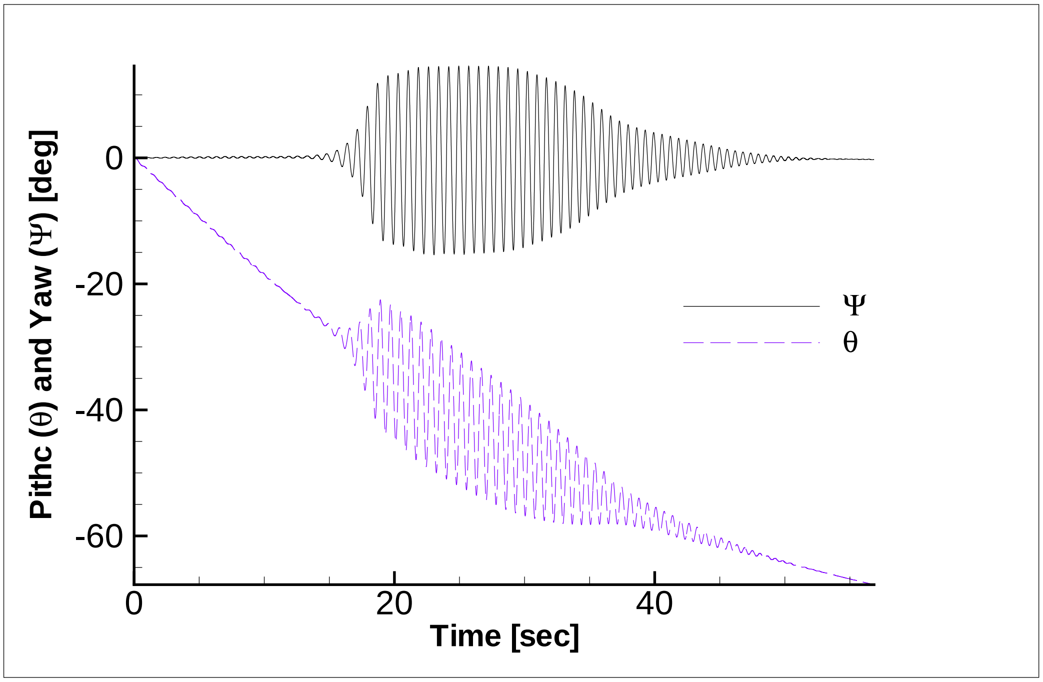

Figure 41: Extended Orientation History of MK-84 AIR at Mach 0.9 at 10,000 ft

the benefit of the ability to indefinitely simulate a store in flight is demonstrated by this event.

4.3 Generic Store Results

To verify the correct effect of moving components on the store motion while using the modified code, another comparison test case is used. The initial conditions of Test Case 1 (M=0.6 at 20,000 ft) are used in this case to first obtain the static solution around the generic store body. The coefficients of force and moments are analyzed to determine the convergence of the flow field around this body. These coefficients were found to reach a steady state solution in 350 iterations, shown in Appendix A.4.

Table 16: Standard Deviations of Forces and Moments in the Final 100 iteration of Generic Store Static Solution

	Fx	**Fy**	**Fz**	**Mx**	**My**	**Mz**
Std. Deviation:	1.44E-4	4.74E-6	5.25E-6	0.00	4.84E-6	2.40E-6

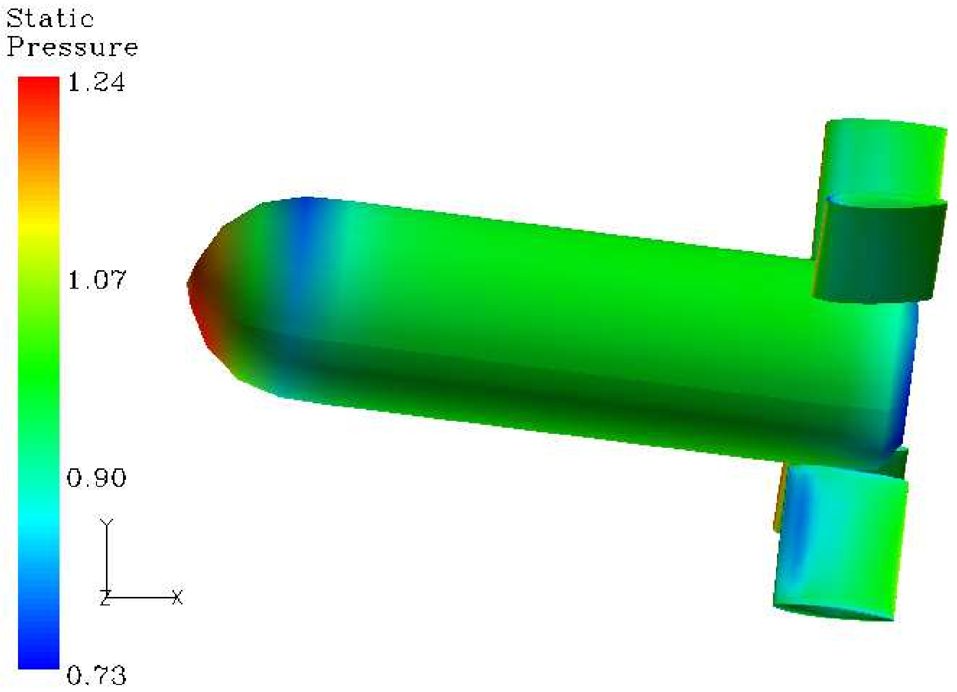

Figure 42: Contours of Static Pressure over the Generic Store in Initial Pitch at Mach 0.6 at 20,000 ft

The standard deviations over the final 100 iterations are given in Table 16. The low standard deviations indicate that the solution is sufficiently converged and that the dynamic simulation may begin.

The prescribed pitching motion is applied over the first 0.6 seconds using the upper fins only, after which all fins remain fixed relative to the store body. Because of its low mass and high drag, the generic store quickly travels backwards in the translating case. The simulation can only be run for 1.4 seconds before the body moves too close to a boundary for the solution to continue without error. However, this is enough time to observe the prescribed motion and the initial response.

The store reaches a maximum attitude of 7.3° nose up in the initial pitching motion, shown in Figure 42. The pitch rate begins to steady out after the fin deflection is removed and is accompanied by the damping of the actual pitch angle. The generic store has a strong tendency to roll, rolling left 30° by the end of the solution. The store also experiences a slight yaw rate which damps out over the simulation. The final yaw angle remains close to zero. The cause of the high roll rate of the generic store in the "X" configuration is unknown. However, because similar roll rates are seen in both solution methods, the probable cause is the geometry and not the solution method. Figures 43 to 48 show the time histories of the pertinent data and the differences at the end of the solution are tabulated in Tables 17 to 19.

The histories of the force and moment coefficients are shown in Figures 43 and 44. The greatest difference between methods seems to appear in the values of C_{Fx}. However, when considering the percent difference, the values of C_{Fx} vary by less than 10% the entire solution. At the end of the solution, the greatest differences are seen in the values of C_{Fz}, C_{Mx}, and C_{My}. While these percent differences are large, the values of these coefficients are near zero for both simulation methods.

The velocities over the solution are shown in Figures 45 and 46. The pitch rate, which is the point of interest because of the prescribed pitch up motion, agrees very

Table 17: Generic Store Body: Forces and Moments after 1.4 seconds

	Force Coefficient [-]			Moment Coefficient [-]		
	x	y	z	x	y	z
Reference Case:	0.0745	0.0057	-4.00E-05	6.00E-05	-0.0008	0.0004
Pinned Case:	0.0793	0.0060	-7.00E-05	5.00E-05	-0.0008	-0.0009
Difference (%):	6.18	5.50	54.55	18.18	8.70	200.00

Table 18: Generic Store Body: Test Velocities after 1.4 seconds

	Velocity [ft/s]			Angular Velocity [rad/s]		
	u	v	w	$\dot{\phi}$	$\dot{\theta}$	$\dot{\psi}$
Reference Case:	13.1663	-43.4651	-3.08E-02	0.3391	0.0321	0.0244
Pinned Case:	13.9865	-43.4421	-4.59E-02	0.4219	0.0329	0.0222
Difference (%):	6.04	0.05	39.44	21.76	2.37	9.70

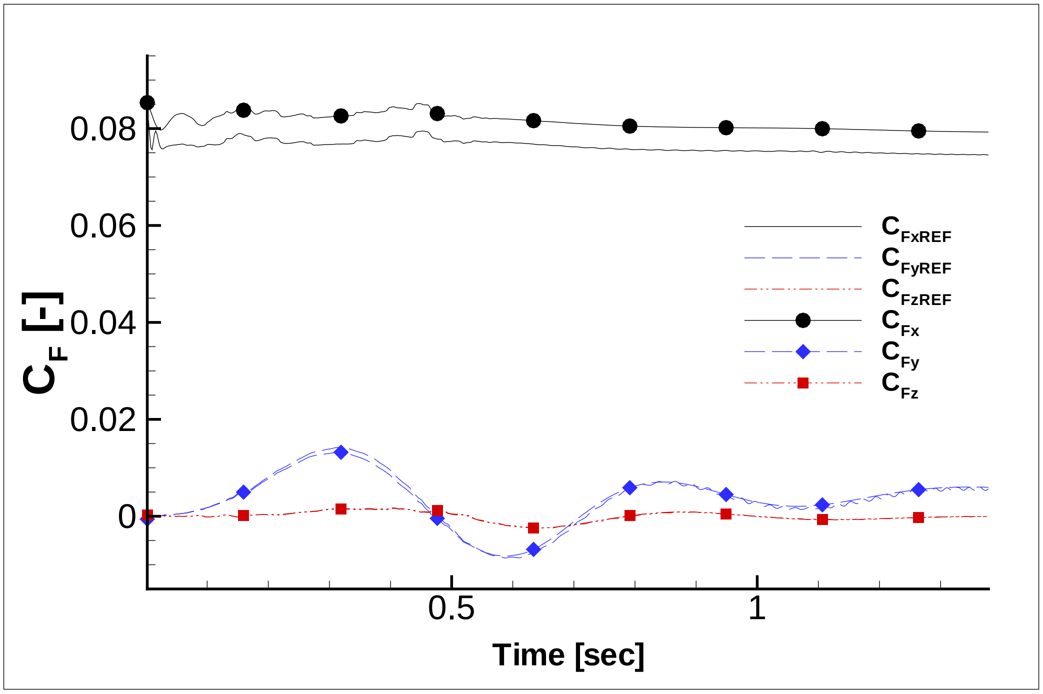

Figure 43: Force History on Generic Store Body at Mach 0.6 at 20,000 ft

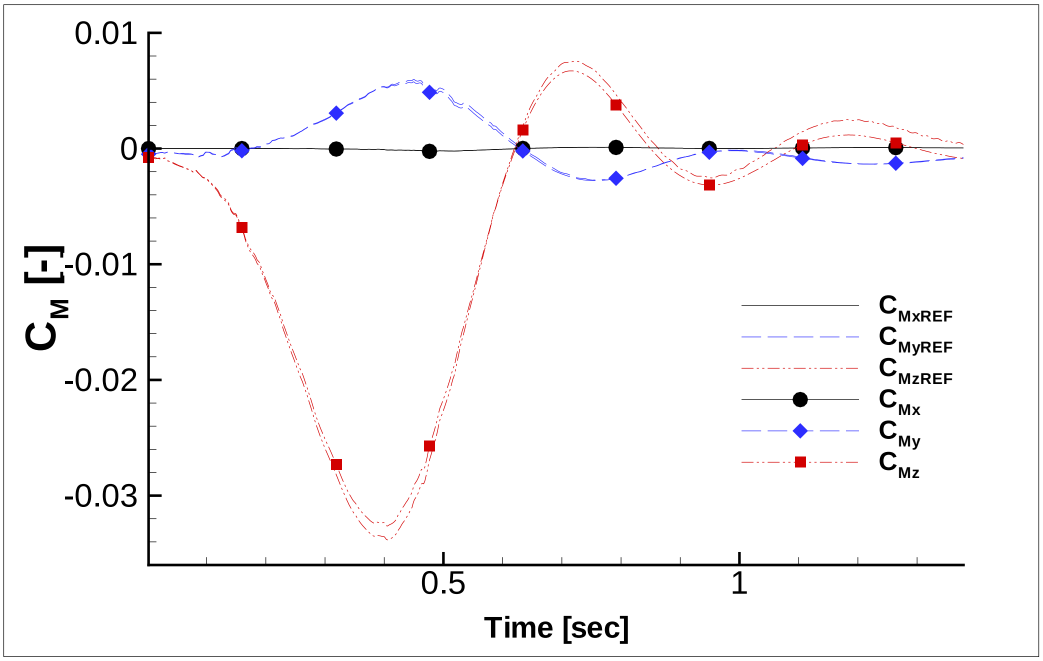

Figure 44: Moment History on Generic Store Body at Mach 0.6 at 20,000 ft

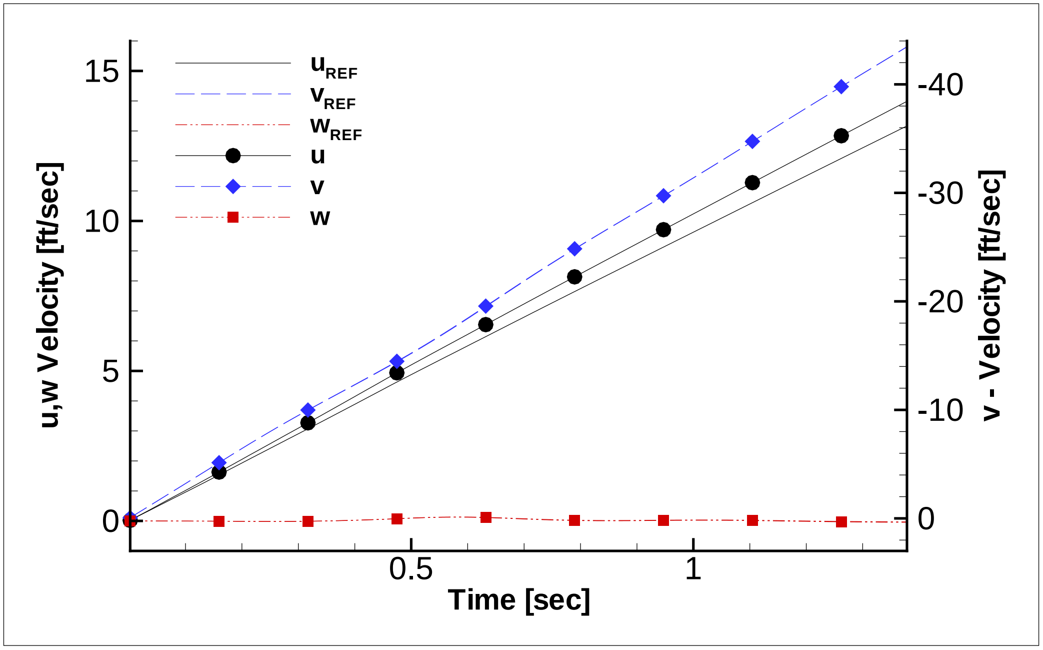

Figure 45: Generic Store Body Velocity History at Mach 0.6 at 20,000 ft

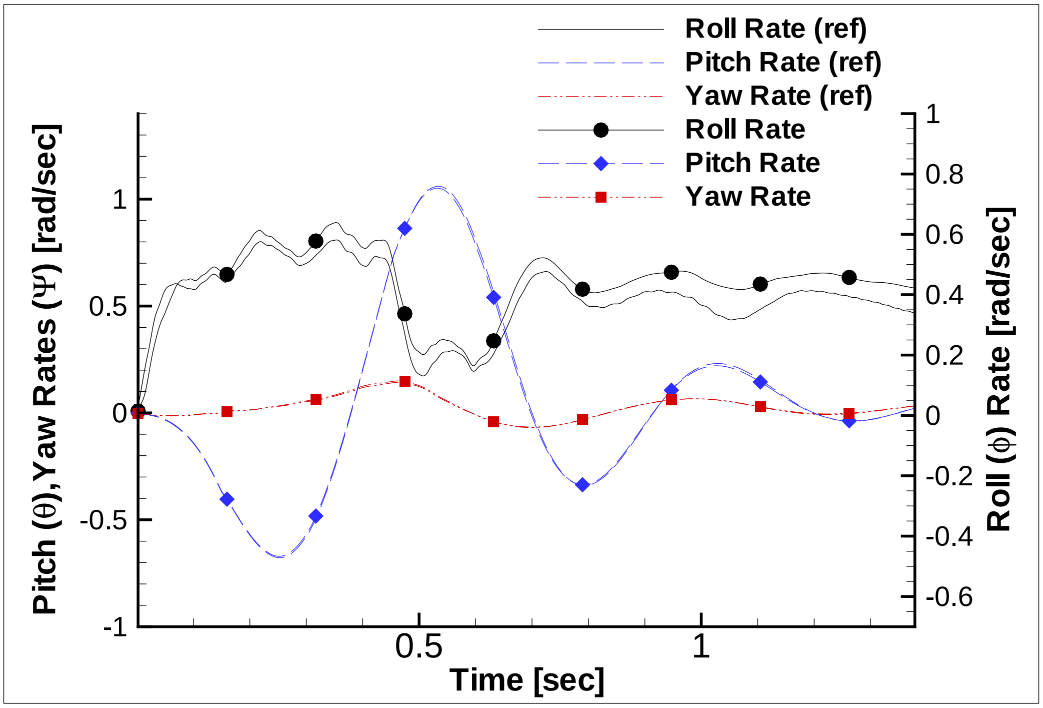

Figure 46: Generic Store Body Angular Velocity History at Mach 0.6 at 20,000 ft

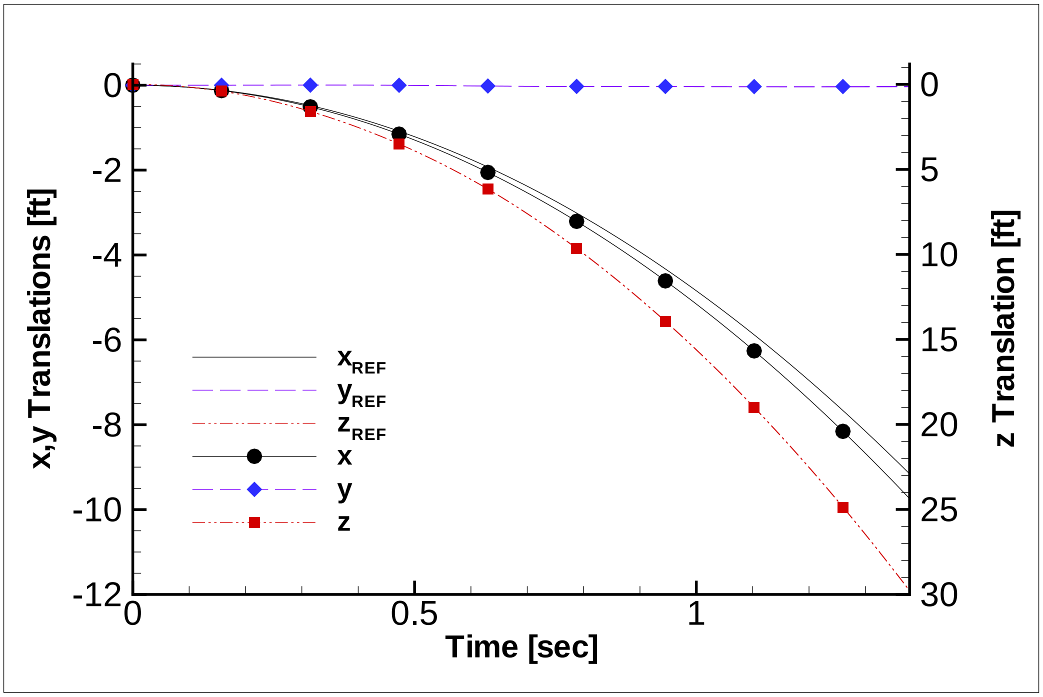

Figure 47: Generic Store Body Trajectory at Mach 0.6 at 20,000 ft

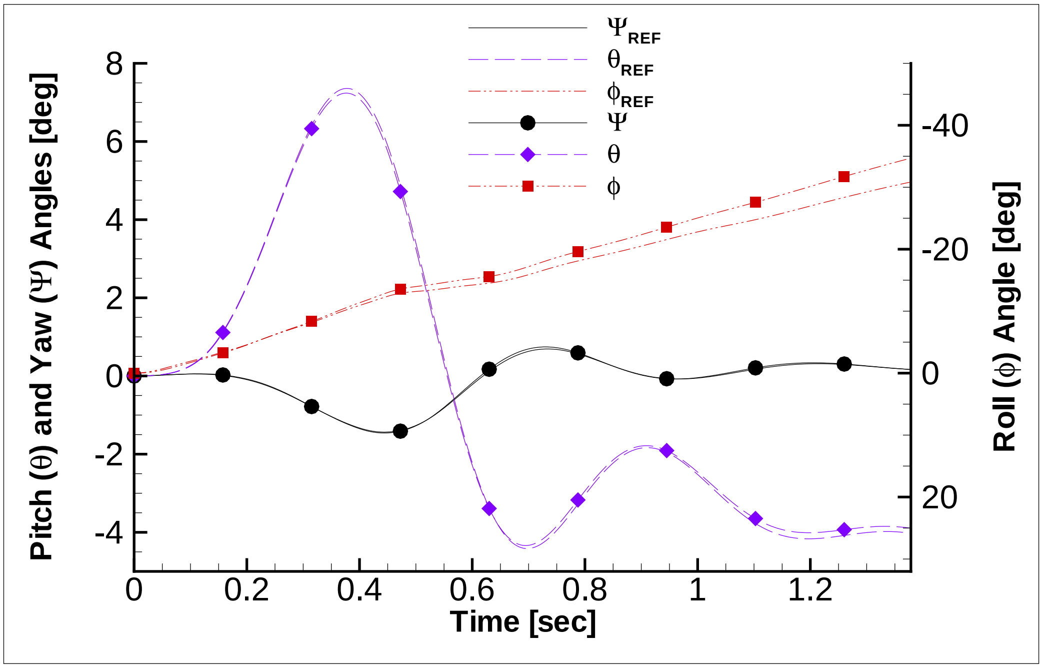

Figure 48: Generic Store Body Orientation History at Mach 0.6 at 20,000 ft

Table 19: Generic Store Body: Trajectory and Orientation after 1.4 seconds

	Translation [ft]			Orientation [degrees]		
	x	y	z	ψ	θ	ϕ
Reference Case:	-9.1565	-0.0364	29.7614	0.16547	-4.0257	-30.8461
Pinned Case:	-9.7397	-0.0257	29.7878	0.16736	-3.8951	-34.6990
Difference (%):	6.17	34.36	0.09	1.14	3.30	11.76

well throughout the solution. The final difference in pitch rate after 1.4 seconds is only 2.4%. The yaw rate, which also oscillates in response to the applied motion, shows 9.7% error at the end of the solution. Most of the differences between the other velocities remain small at the end of the solution, with the exception of w-velocity and the roll rate. The w-velocities from both methods have values near zero. The roll rate, however, is not close to zero. There is less agreement in roll rate between the two methods than the other angular velocities, although it follows the same trend in each method. Grid interpolation differences in the region of the moving components may contribute to these differences.

The trajectory and orientation histories of the store are shown in Figures 47 and 48. At the end of the solution, close agreement is seen between solutions (Table 19). The difference of the vertical translation of the store is less than 1%, and the difference in the final pitch angle is only 3.3%. The greatest difference is in the y-translations, which are both close to zero because of the low yaw rate of the store. While the roll rate shows slight differences through the solution, the difference in the actual roll angle grows slowly to a maximum of 11.8% at the end of the solution.

Overall, the two methods show close agreement in this model. The difference between methods grows as time progresses, as seen in the MK-84 AIR testing. These differences may be attributed to the grid interpolation errors, which increase as the store gains velocity and larger interpolation jumps are required between time steps.

Such errors may be smaller in the pinned case because the dynamic group does not translate.

Because of the success of the comparison tests, the pinned solution was continued until 4 seconds of solution time was reached. Figures 49 to 54 show the history of the generic store over this time. This solution captures the entire response of the store to the initial pitching motion, something the unmodified Beggar code was unable to do. The store falls 250 feet in this time. As the store rotates nose down, C_{Fy} increases as C_{Fx} decreases. The forces and moments damp out within 2 seconds, as do the pitch and yaw rates. The simulation ends with the store in a final orientation of 12° nose down, having rolled left 87°. The final yaw angle remains close to zero. Although this simulation is relatively short compared to the length of the extended MK-84 AIR simulations, it shows the potential usefulness of these modifications when predicting the dynamic stability of a store.

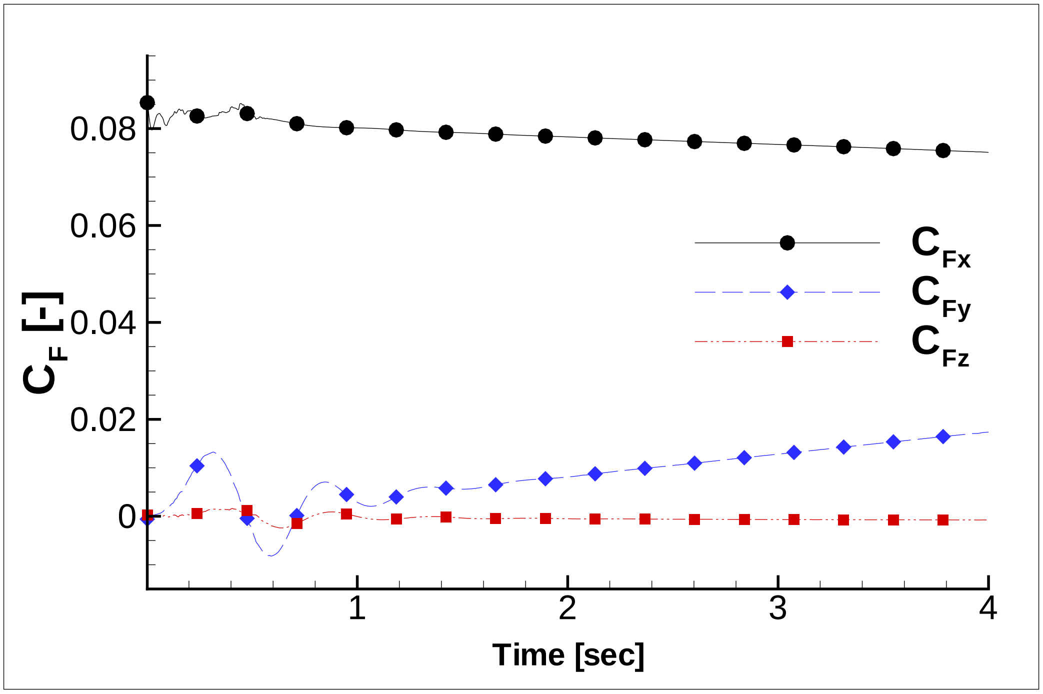

Figure 49: Extended Force History on Generic Store Body at Mach 0.6 at 20,000 ft

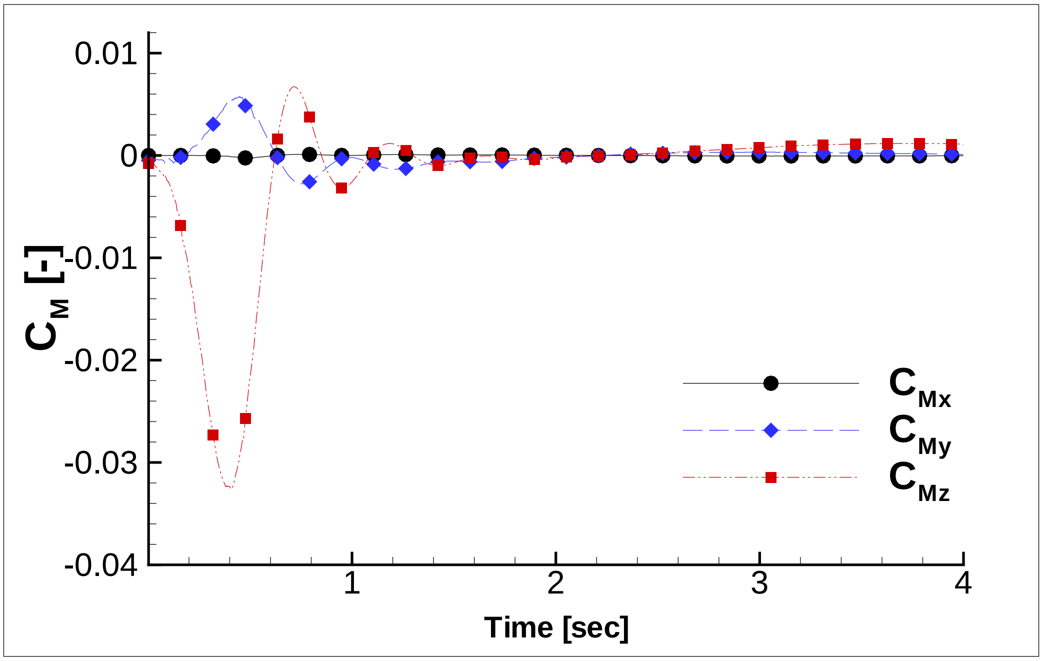

Figure 50: Extended Moment History on Generic Store Body at Mach 0.6 at 20,000 ft

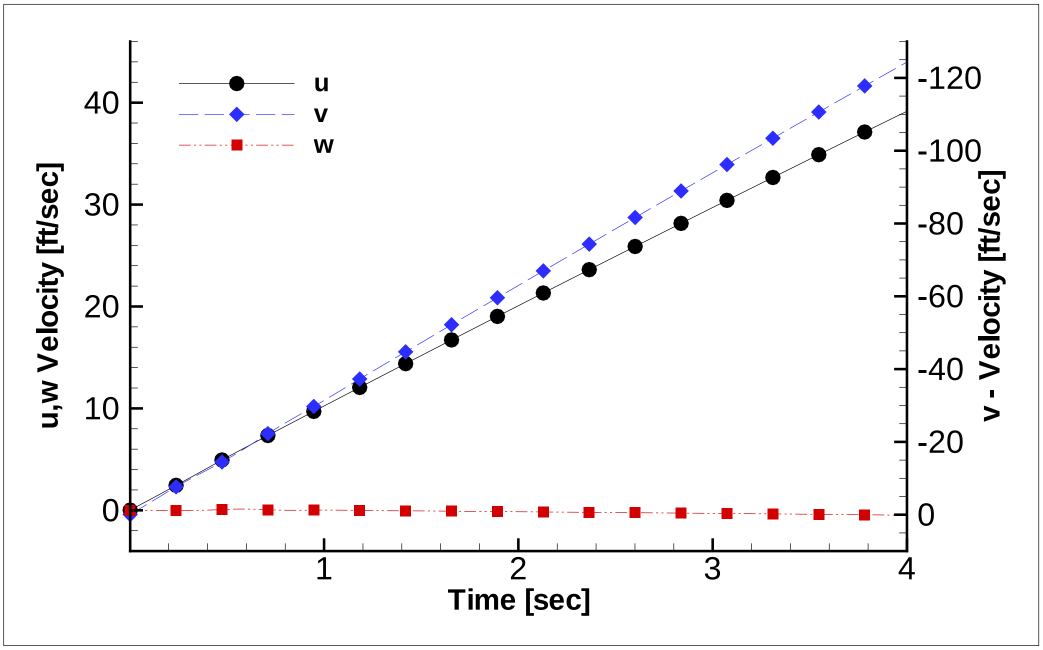

Figure 51: Extended Velocity History of Generic Store Body at Mach 0.6 at 20,000 ft

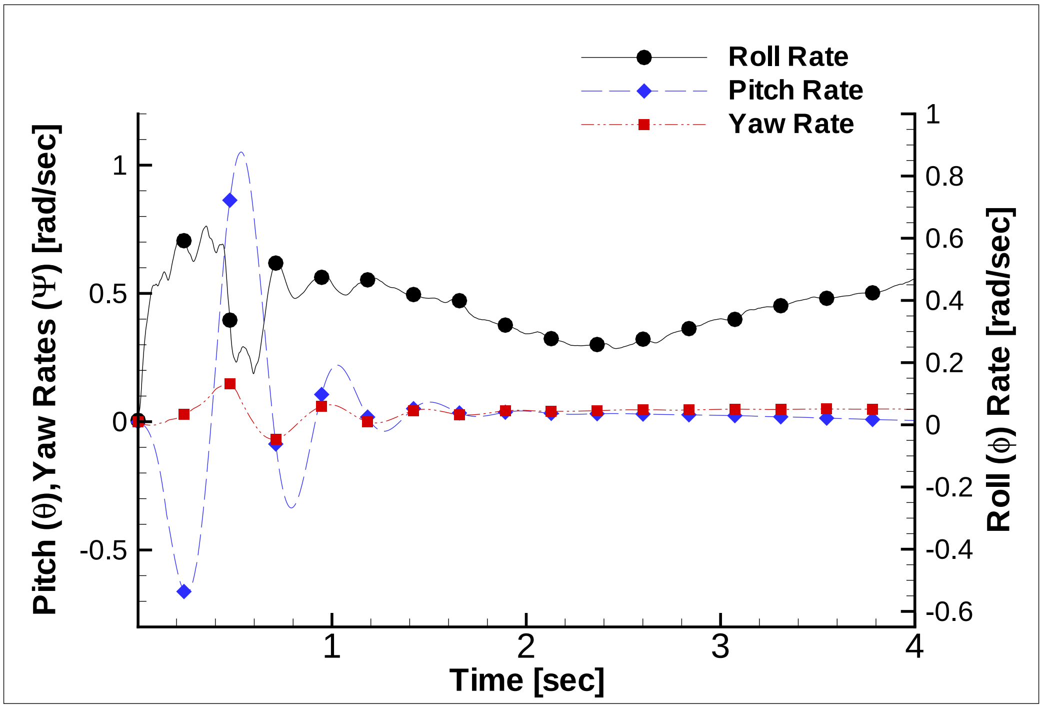

Figure 52: Extended Angular Velocity History of Generic Store Body at Mach 0.6 at 20,000 ft

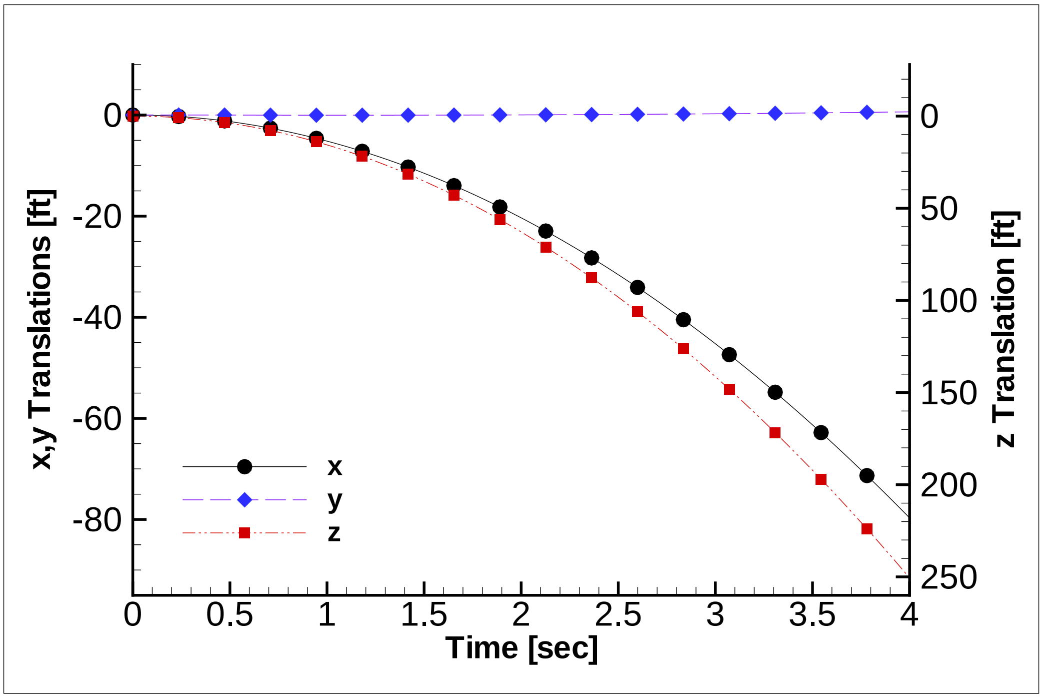

Figure 53: Extended Trajectory of Generic Store Body at Mach 0.6 at 20,000 ft

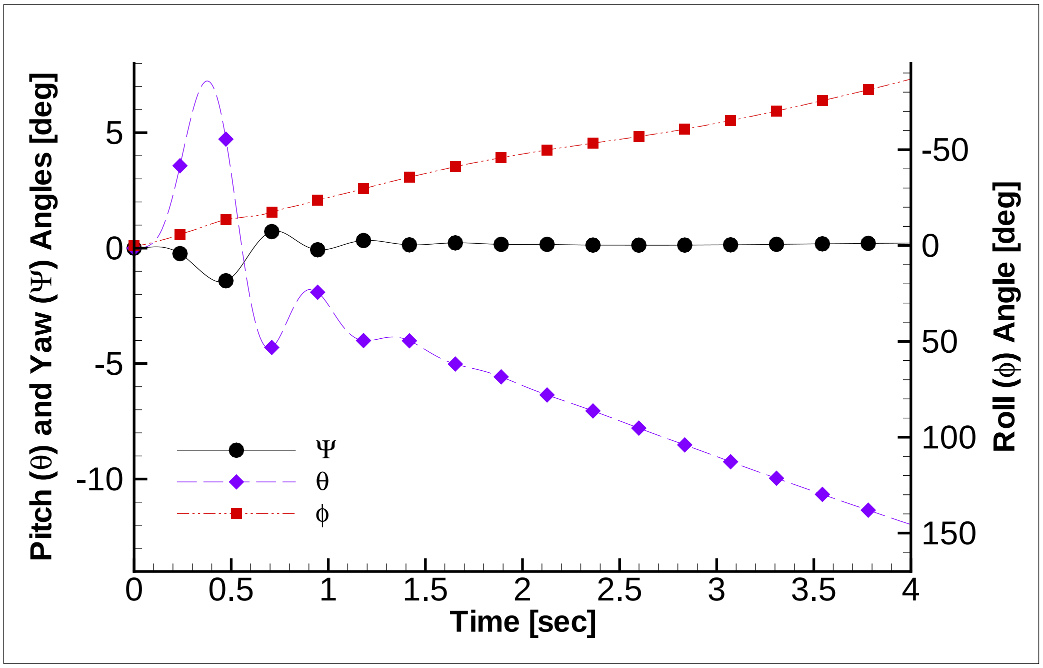

Figure 54: Extended Orientation History of Generic Store Body at Mach 0.6 at 20,000 ft

V. Conclusions

Beggar, the premier CFD code developed by and for the United States Air Force, has been successfully modified to enable the extended simulation of a store in free flight. The implementation of mesh motion on the background mesh was confirmed to be successful through the use of a supersonic compression ramp. Testing used different combinations of flow and mesh motion, which all resulted in the same flow solution over the ramp. Furthermore, the flow visualization corrections, designed to include that mesh motion, were found to work correctly through these tests.

The MK-84 AIR comparison tests found good agreement between the modified and unmodified code for the two test cases used. This confirms the success of the pinned method which removes the requirement for an inertially-fixed background mesh. Additionally, two extended simulations of the MK-84 AIR were run. The long-term dynamic behavior of the store was observed, demonstrating the strength of these modifications. These simulations (56 and 135 seconds in length) would have been intractable by the Beggar code in the past. For a short term simulation, there is an impressive reduction in computational cost using the pinned method with a smaller background mesh.

After the successful removal of the inertial background mesh, the generic store tests verified the pinned solution with moving components present that applied a pitch up motion to the store. The motion was found to closely agree between the translating and pinned methods. The original, translating method of free flight simulation could not totally capture the store's dynamic response to the initial pitch. However, with the current modifications, a extended simulation adequately captured the entire long-term dynamic response of the store to this prescribed motion.

The errors in all comparison tests were observed to increase with time. The probable cause of this is the grid interpolation of the store. As the store gains velocity, it takes larger interpolation jumps and results in increased error. These interpolation jumps are larger in the translating case because the interpolations in the pinned case occur in response to rotation only. Therefore, the pinned solution may actually contain less grid interpolation error than the translating solution.

5.1 Future Research

While this research has shown that indefinite free flight simulations with Beggar are possible, more work needs to be done to implement this for all possible scenarios. Beggar has the ability to simulate many different types of prescribed motion and constraint forces in single and multi-body problems. How the current modifications affect these additional constraints and motion options is unknown. In addition, these modifications have not been applied to the moving chute or kone capabilities of Beggar. For these cases, further modifications must be accomplished.

Currently, atmospheric reference values are taken from the initial conditions input by the user in the form of density and speed of sound. Beggar holds these values constant and uses them throughout the simulation, even as the store falls thousands of feet. In reality, these values obviously change as the store falls. Because of this, the accuracy of any extended free flight simulation would benefit from the addition of an atmospheric model to Beggar.

These modifications could be employed to predict the dynamic stability of a store. This would be an invaluable extension of the Beggar code, providing increased flexibility and capacity for weapons certification. Additionally, the removal of the inertial background mesh introduces the potential capability to simulate store separation events from maneuvering aircraft. However, further research and adaptations to the Beggar code must first be accomplished.

These results have confirmed the success of removing the inertial background mesh requirement for the single and multi-body problems. This has enabled the previously impossible simulation of a store in indefinite free flight using Beggar. This research has expanded Beggar's capability and will allow it to continue to meet the growing weapons certification demands of the United States Air Force.

Appendix A. Extended Results

Supporting figures for each of the three computational test models are given here.

A.1 Compression Ramp Case Results

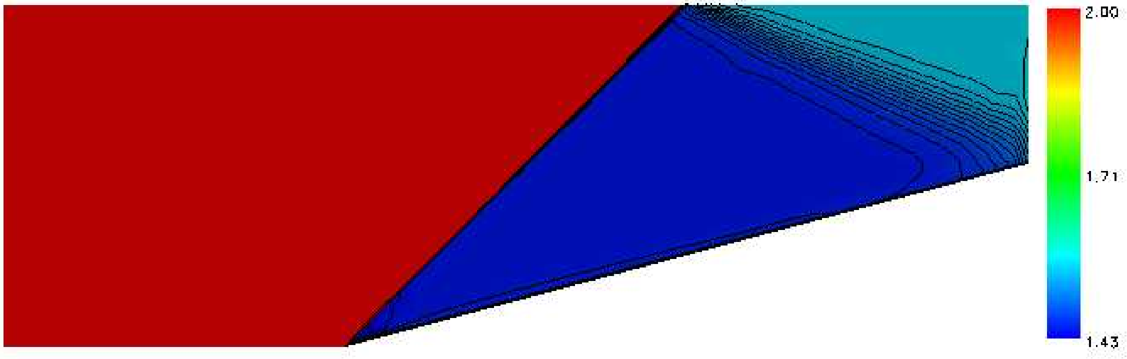

Figure A.1: Ramp Case 2: Mach 1.5 flow and Mach 0.5 mesh motion

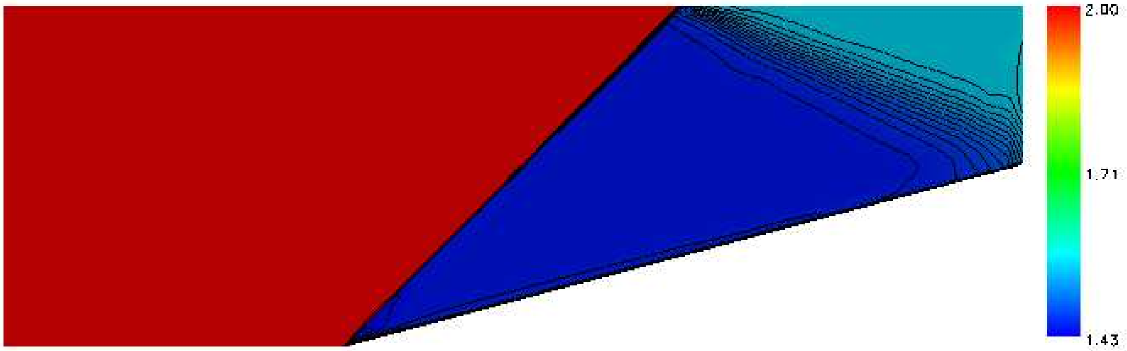

Figure A.2: Ramp Case 3: Mach 0.5 flow and Mach 1.5 mesh motion

A.2 MK-84 AIR Convergence History

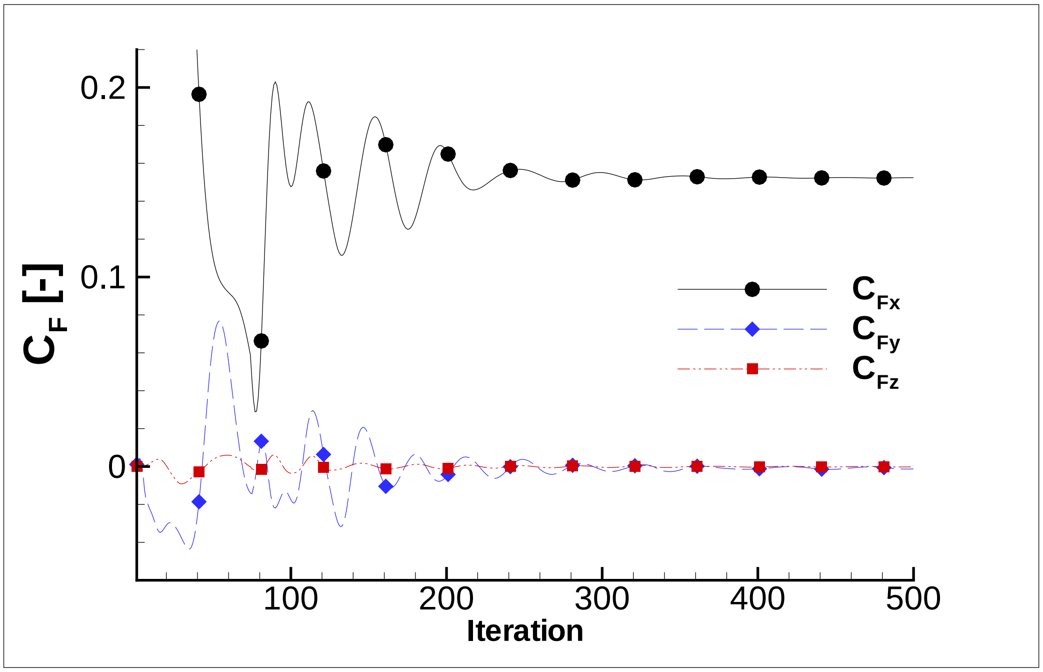

Figure A.3: Convergence of Static Forces on MK-84 AIR

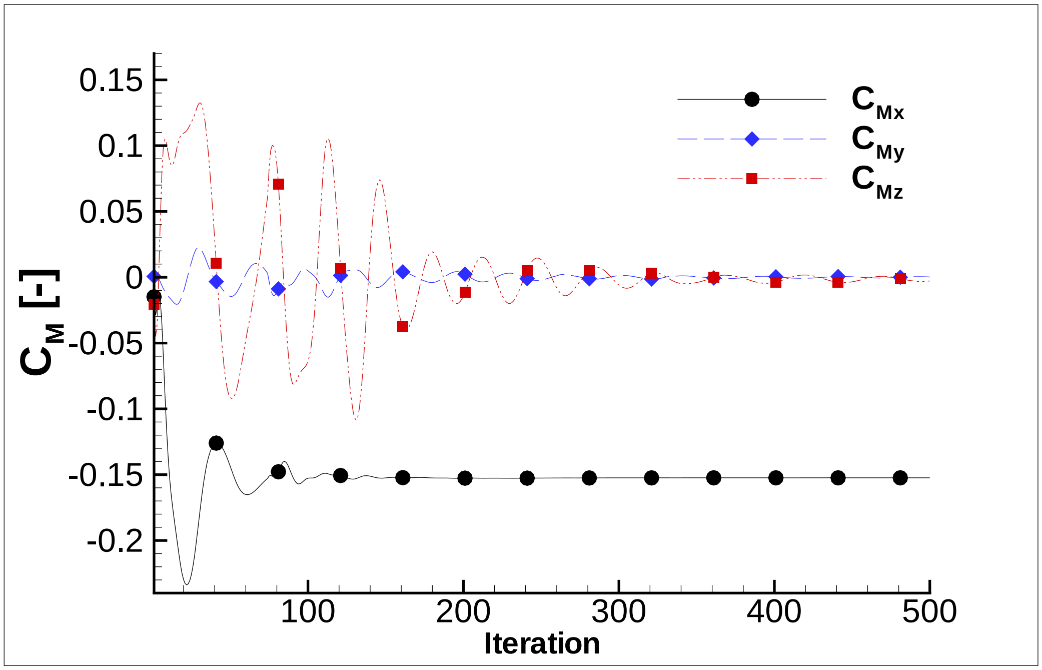

Figure A.4: Convergence of Static Moments on MK-84 AIR

A.3 MK-84 AIR Comparison Test Case 2

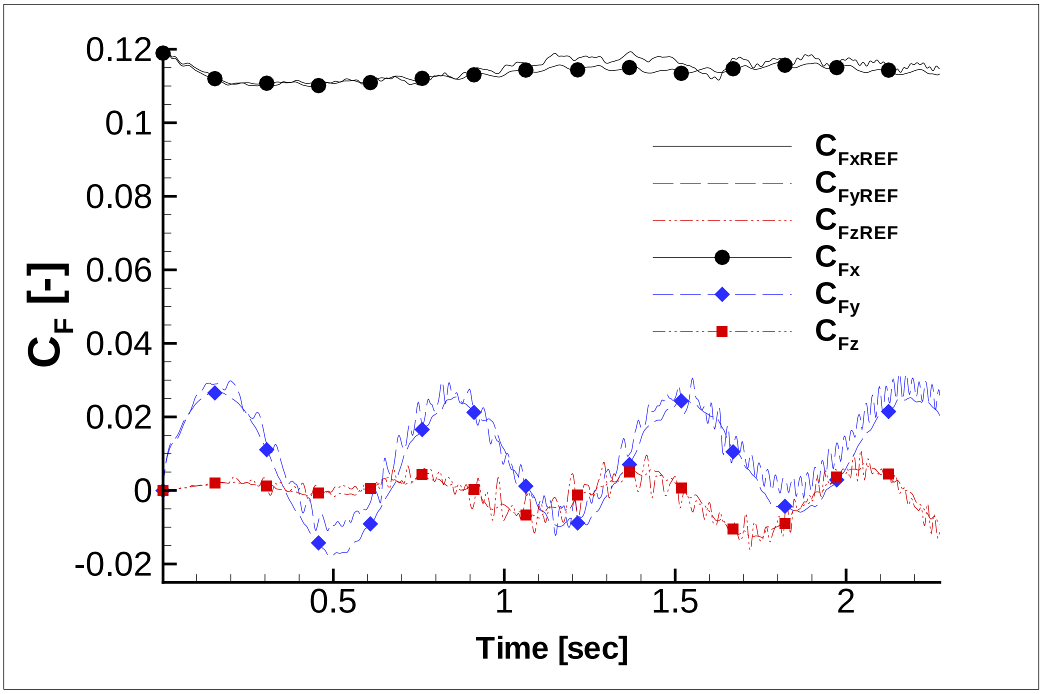

Figure A.5: Force History on MK-84 AIR body at Mach 0.9 at 10,000 ft

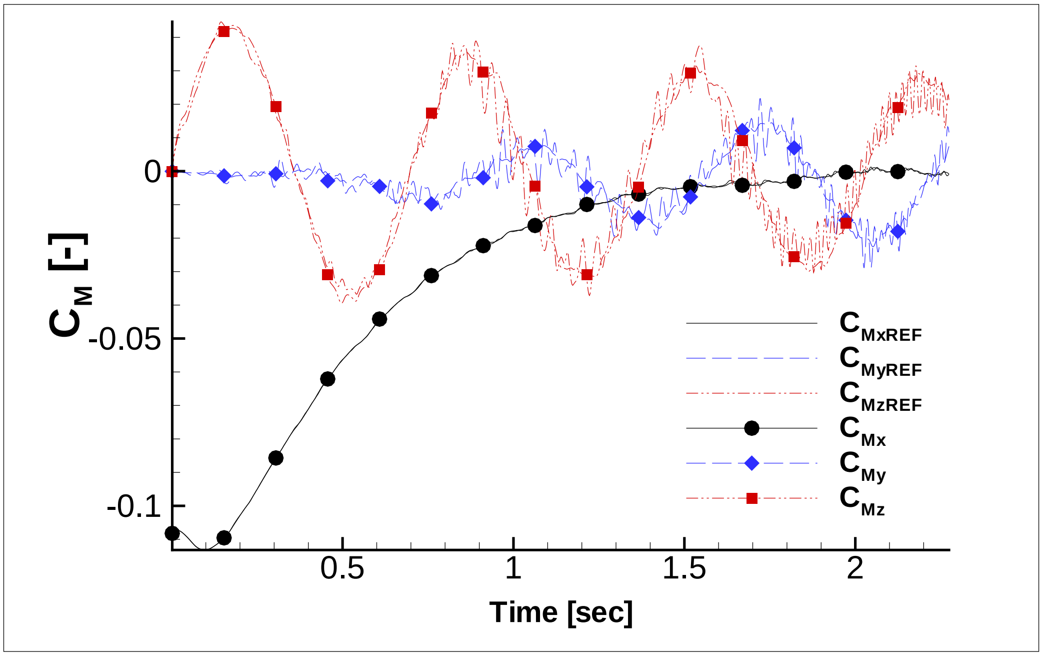

Figure A.6: Moment History on MK-84 AIR body at Mach 0.9 at 10,000 ft

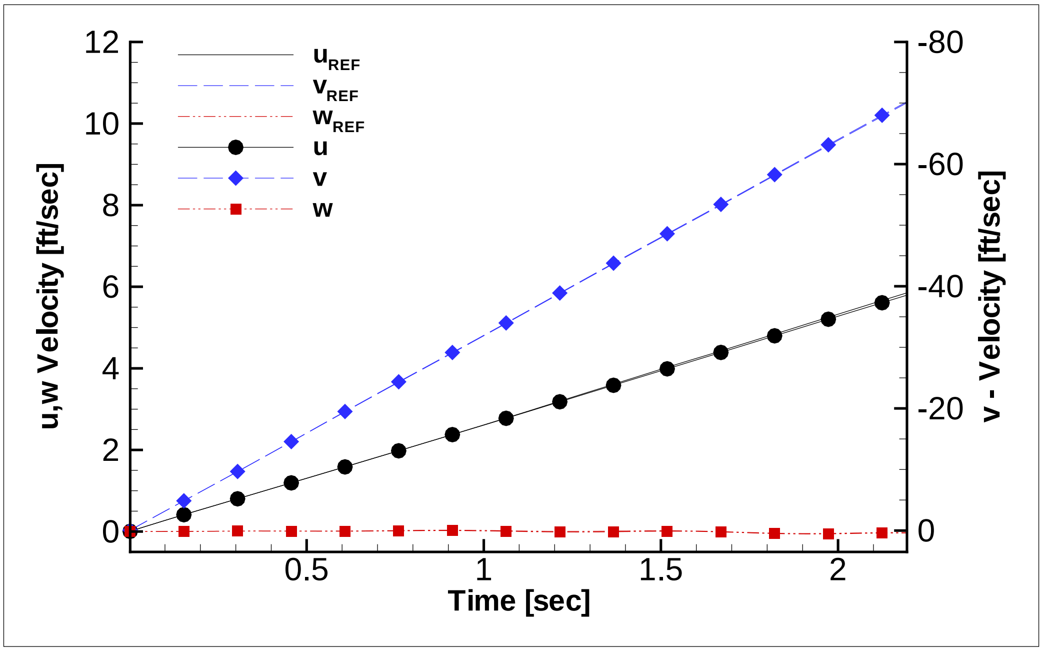

Figure A.7: MK-84 AIR Velocity History at Mach 0.9 at 10,000 ft

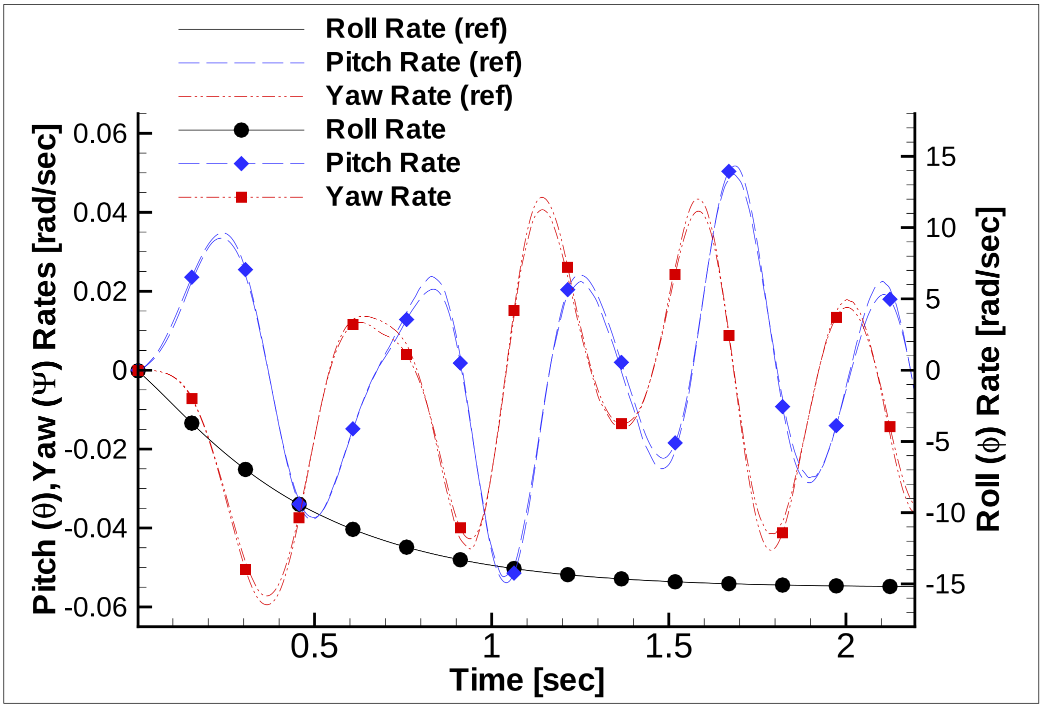

Figure A.8: MK-84 AIR Angular Velocity History at Mach 0.9 at 10,000 ft

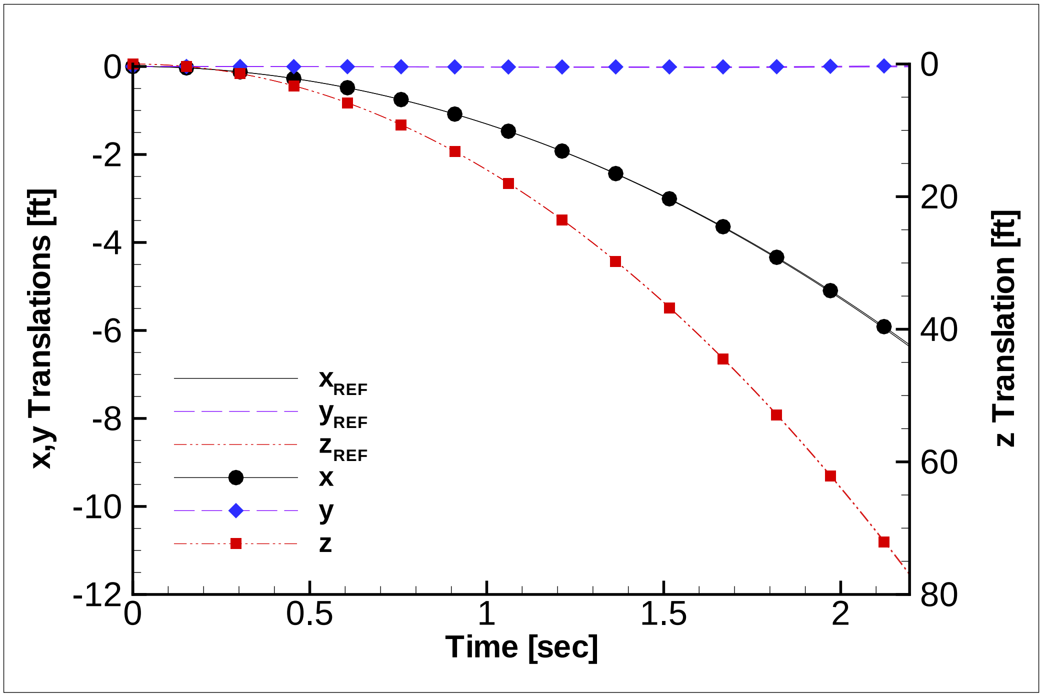

Figure A.9: MK-84 AIR Trajectory at Mach 0.9 at 10,000 ft

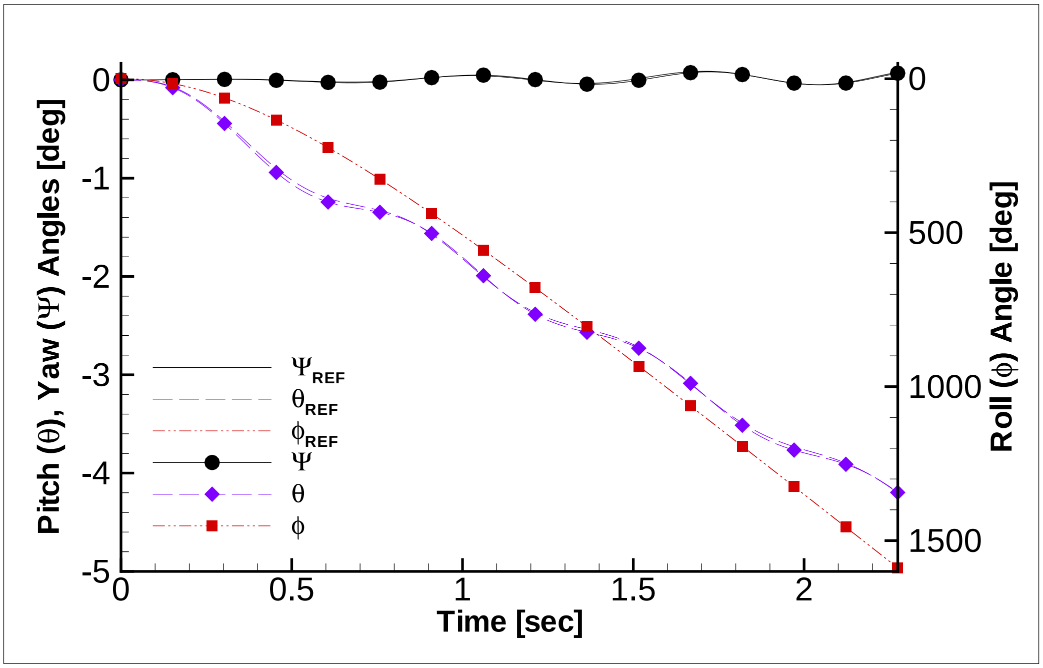

Figure A.10: MK-84 AIR Orientation at Mach 0.9 at 10,000 ft

A.4 Generic Store Body Convergence History

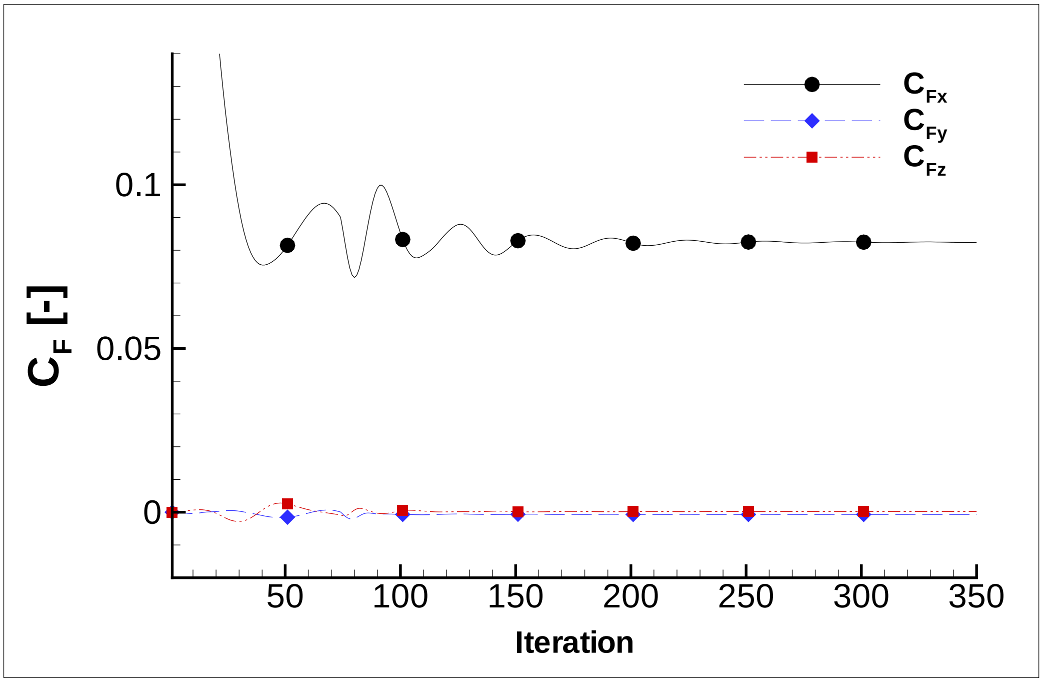

Figure A.11: Convergence of Static Forces on Generic Store Body

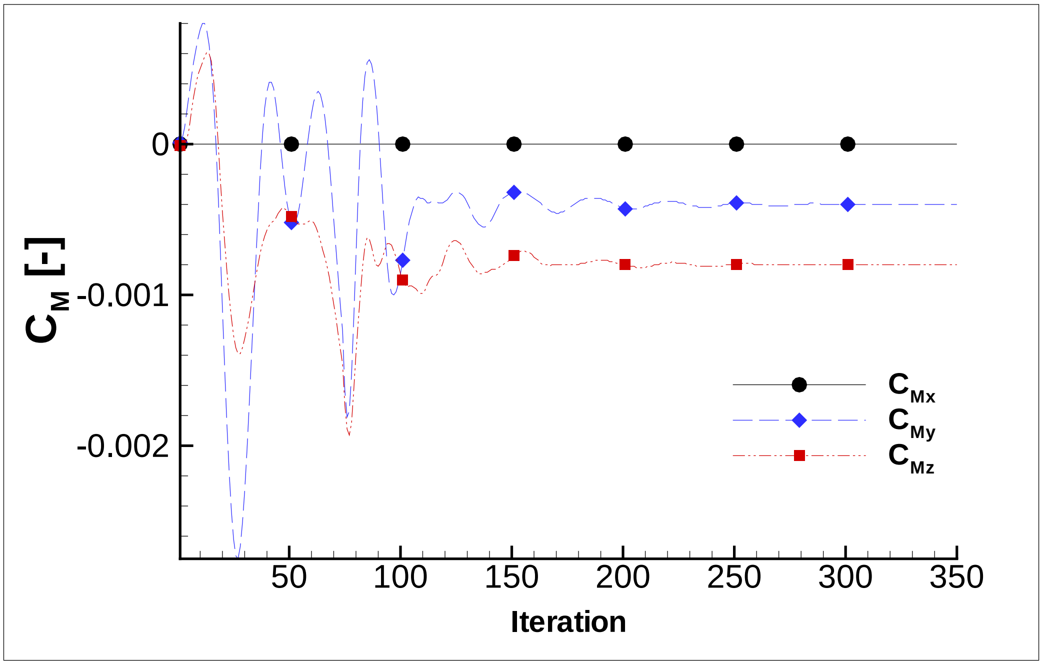

Figure A.12: Convergence of Static Moments on Generic Store Body

Appendix B. Beggar Input Files

The various input files needed for the MK-84 AIR and generic store body simulations are provided in this Appendix.

B.1 MK-84 AIR Input File

```
# -------------------------------(MK-84 AIR TESTING)--------------------------------
#----------------------------------------------------------------------------------
#                               INITALIZATION PARAMETERS
#----------------------------------------------------------------------------------

verbose = 3
mach = 0.60
ptol = 1e-5
nopatch
dt= 10.0

#----------------------------------------------------------------------------------
#                                  SIX-DOF PARAMETERS
#----------------------------------------------------------------------------------

sixdof gravity = <0.0,-32.18,0>            # ft/s^2   for AOA = 0 deg
sixdof density = 0.001262                  # Slug/ft^3  @ 20k ft
sixdof soundspd =1037.                     # ft/s       @ 20k ft
sixdof refl = 0.083333                     # Ref length conversion

#----------------------------------------------------------------------------------
#                               FLOW SOLVER PARAMETERS
#----------------------------------------------------------------------------------

inner = 40
inner_tol=1.e-8
inner_tol_ratio=1.e+4
stencil=inviscid2
solver=second order, full, euler, steg_warm_xair jacobians,
        implicit bcupdate, primitive extrap,steger_warming right_side
limiter = vanalbada

#------------------------------------------------------------------------------
#                                GRID ASSEMBLY
#------------------------------------------------------------------------------

readgrids '../grid/cube.grd' as plot3d ascii

readgrids '../grid/cart.p3ds' as plot3d binary
  tag 'mk84air_cart_grid_SB'

include '../beg/mk84airinv.beg'

#------------------------------------------------------------------------------
#                             GRID INITIALIZATION
#------------------------------------------------------------------------------

sb 1
init from 'mk84_restart' 1

sb 2
init from 'mk84_restart' 2

sb 3
init from 'mk84_restart' 3

#------------------------------------------------------------------------------
#                            FORCE SPECIFICATIONS
#------------------------------------------------------------------------------

forcespec "mk84air_tot_fspec": dump every 1 to "store_forces.dat"
with noheader
 with refl=---   #Approx diameter of store
 with refa=---   #Approx cross sectional area using body diameter
 with mcenter = <---,---,--->
forcespec "mk84air_tot_fspec": add "mk84air_body","mk84air_fin"

#-----------------------------------------------------------------------------
#                    INERTIAL AND DYNAMIC SPECIFICATIONS
#-----------------------------------------------------------------------------

dynamicspec "mk84air_body_ds":
             add sb 'mk84air_inv';
             add sb 'mk84air_cart_grid_SB';
             add fspec 'mk84air_tot_fspec';
             mass = 62.7;     # Slug
             ixx = ---;       # Slug ft^2
             iyy = ---;       # Slug ft^2
             izz = ---;       # Slug ft^2
             ixy = ---; iyz = ---; ixz = ---;
             cg = <---,---,--->;      # feet!
             trelease = 0;
             dump idaps left release;
             dump gandc z down

#-----------------------------------------------------------------------------
#                     SET WORLDSIDE FOR CCUT OPTIONS
#-----------------------------------------------------------------------------

sb 1
 g 1
  set (1,1,1) (2,2,2) to worldside
```

B.2 MK-84 AIR Boundary Conditions

The configuration file “mk84airinv.beg” containing the surface boundary conditions of the MK-84 AIR is shown here:

```
readgrids ’../grid/mk84_new.grd’ as plot3d ascii
  tag ’mk84air_inv’
  g 1
    set ”mk84air_body”  =  (1,1,1) (*,*,1) to tangent
    set ”mk84air_fin”   =  (43,1,1) (69,1,29) to tangent
    set ”mk84air_fin”  +=  (43,*,1) (69,*,29) to tangent
  g 2
    set ”mk84air_body” +=  (1,1,1) (*,*,1) to tangent
    set ”mk84air_fin”  +=  (43,1,1) (69,1,29) to tangent
    set ”mk84air_fin”  +=  (43,*,1) (69,*,29) to tangent
  g 3
    set ”mk84air_body” +=  (1,1,1) (*,*,1) to tangent
    set ”mk84air_fin”  +=  (43,1,1) (69,1,29) to tangent
    set ”mk84air_fin”  +=  (43,*,1) (69,*,29) to tangent
  g 4
    set ”mk84air_body” +=  (1,1,1) (*,*,1) to tangent
    set ”mk84air_fin”  +=  (43,1,1) (69,1,29) to tangent
    set ”mk84air_fin”  +=  (43,*,1) (69,*,29) to tangent
```

B.3 MK-84 AIR Time Step Ramping Schedule

The time step and Newton iteration ramping schedule are shown here. The time steps are input as non-dimensional time, and correspond to different physical time steps depending on the initial conditions of the case. The physical time step for the 20,000 ft cases are shown.

```
apply_at_iter=1
  dtiter=1
  dt=4.5      # 0.000361469 seconds @ 20k ft
```

```
apply_at_iter=25
  dt=14.5   # 0.001164735 seconds @ 20k ft
apply_at_iter=75
  dtiter=3
  dt=24.5   # 0.001968812 seconds @ 20k ft
```

B.4 Generic Store Body Specifications

The force and dynamic specifications used in the input file for the generic store body problem are shown here. As seen, each SMC and the SMB require their own force and dynamic specification. The SMB dynamic spec is immediately followed by the dynamic specs of the four fins. The auxiliary input file with the additional prescribed motion cycle designed to rotate the fins back to zero is also shown.

```
#---------------------------------------------------------------------------
#                          FORCE SPECIFICATIONS
#---------------------------------------------------------------------------

forcespec ”fin1fs”: add ”fin1” forcespec ”fin1fs”: dump every 1 to
”fin1.force”
            with refl=1.5       # Grid units (length of fin)
            with refa=3.17      # Grid units (fin area)
            with mcenter=<0.0,4.5,0.0> # mcenter Local grid units

forcespec ”fin2fs”: add ”fin2” forcespec ”fin2fs”: dump every 1 to
”fin2.force”
            with refl=1.5       # Grid units (length of fin)
            with refa=3.17      # Grid units (fin area)
            with mcenter=<0.0,4.5,0.0> # mcenter Local grid units

forcespec ”fin3fs”: add ”fin3” forcespec ”fin3fs”: dump every 1 to
”fin3.force”
            with refl=1.5       # Grid units (length of fin)
            with refa=3.17      # Grid units (fin area)
```

```
           with mcenter=<0.0,4.5,0.0> # mcenter Local grid units

forcespec ”fin4fs”: add ”fin4” forcespec ”fin4fs”: dump every 1 to
”fin4.force”
           with refl=1.5      # Grid units (length of fin)
           with refa=3.17     # Grid units (fin area)
           with mcenter=<0.0,4.5,0.0> # mcenter Local grid units

forcespec ”bodyfs”: add ”body” forcespec ”bodyfs”: dump every 1 to
”body.force”
           with refl=2.26     # Grid units ( diameter of body)
           with refa=63.314   # Grid units ( 2 pi r h)
           with mcenter=<4.45,0.0,0.0> # mcenter Local grid units

#------------------------------------------------------------------------
#                  INERTIAL AND DYNAMIC SPECIFICATIONS
#------------------------------------------------------------------------

dynamicspec ”bodyds”:
    add fspec ’bodyfs’;
    add sb ’body’;
    add sb ’cart_grd_sb’;
    mass = 0.776;    # 25 / 32.2  = slugs
    ixx = 0.00001;   #represents a solid cylinder 1” rad by 9” long
    iyy = 0.0364;    #represents a solid cylinder 1” rad by 9” long
    izz = 0.0364;    #represents a solid cylinder 1” rad by 9” long
    ixy = 0.0; iyz = 0.0; ixz = 0.0;
    cg = <0.333, 0.0, 0.0>;   # in feet NOT inches
    trelease = 0.0;
    dump idaps left release;
    dump gandc z down

dynamicspec ”fin1ds”:
    add sb ’fin1’;
    add fspec ’fin1fs’;
    store moving component with prescribed motion (option1);
    vector along smc axis of rotation = <0.0,0.5,0.5>;
    point on smc axis of rotation = <0.687499725,0.0,0.0>;
    time motion begins = 0.0;
    time acceleration ends = 0.15;
    time deceleration begins = 0.15;
    time motion ends = 0.3;
    maximum angular velocity = 0.9;
    mass = 0.031;      # 1 / 32.2
    ixx = 0.0000832;  # slug ft^2
    iyy = 0.0000403;  # slug ft^2
    izz = 0.0001214;  # slug ft^2
    ixy = 0.0; iyz = 0.0; ixz = 0.0;
    cg = <0.0, 0.04, 0.0>; # in feet NOT inches
    dump idaps left release;
    dump gandc z down;
    trelease = 0.0

dynamicspec ”fin2ds”:
    add sb ’fin2’;
    add fspec ’fin2fs’;

    store moving component with prescribed motion (option1);
    vector along smc axis of rotation = <0.0,-0.5,0.5>;
    point on smc axis of rotation = <0.687499725,0.0,0.0>;
    time motion begins = 50.0;
    time acceleration ends = 50.15;
    time deceleration begins = 50.15;
    time motion ends = 50.3;
    maximum angular velocity = 0.9;
    mass = 0.031;      # 1 / 32.2
    ixx = 0.0000832;  # slug ft^2
    iyy = 0.0000403;  # slug ft^2
    izz = 0.0001214;  # slug ft^2
    ixy = 0.0; iyz = 0.0; ixz = 0.0;
    cg = <0.0, 0.04, 0.0>; # in feet NOT inches
    dump idaps left release;
    dump gandc z down;
    trelease = 0.0

dynamicspec ”fin3ds”:
    add sb ’fin3’;
    add fspec ’fin3fs’;

    store moving component with prescribed motion (option1);
    vector along smc axis of rotation = <0.0,0.5,0.5>;
    point on smc axis of rotation = <0.687499725,0.0,0.0>;
    time motion begins = 50.0;
    time acceleration ends = 50.15;
    time deceleration begins = 50.15;
    time motion ends = 50.3;
    maximum angular velocity = 0.9;
    mass = 0.031;      # 1 / 32.2
    ixx = 0.0000832;  # slug ft^2
    iyy = 0.0000403;  # slug ft^2
    izz = 0.0001214;  # slug ft^2
    ixy = 0.0; iyz = 0.0; ixz = 0.0;
    cg = <0.0, 0.04, 0.0>; # in feet NOT inches
    dump idaps left release;
    dump gandc z down;
    trelease = 0.0

dynamicspec ”fin4ds”:
    add sb ’fin4’;
    add fspec ’fin4fs’;
    store moving component with prescribed motion (option1);
    vector along smc axis of rotation = <0.0,−0.5,0.5>;
    point on smc axis of rotation = <0.687499725,0.0,0.0>;
    time motion begins = 0.0;
    time acceleration ends = 0.15;
    time deceleration begins = 0.15;
    time motion ends = 0.3;
    maximum angular velocity = 0.9;
    mass = 0.031;      # 1 / 32.2
    ixx = 0.0000832;  # slug ft^2
    iyy = 0.0000403;  # slug ft^2
    izz = 0.0001214;  # slug ft^2
    ixy = 0.0; iyz = 0.0; ixz = 0.0;
    cg = <0.0, 0.04, 0.0>; # in feet NOT inches
    dump idaps left release;
    dump gandc z down;
    trelease = 0.0

#--------------------------------------------------------------------------
#                  GSB Auxiliary Input File
#--------------------------------------------------------------------------

override
dynamicspec 'fin1ds':
         additional prescribed motion cycle;
         time motion begins       = 0.304;
         time acceleration ends   = 0.454;
         time deceleration begins = 0.454;
         time motion ends         = 0.604;
         maximum angular velocity =-0.9

dynamicspec 'fin4ds':
         additional prescribed motion cycle;
         time motion begins       = 0.304;
         time acceleration ends   = 0.454;
         time deceleration begins = 0.454;
         time motion ends         = 0.604;
         maximum angular velocity =-0.9
```

Bibliography

1. "Beggar Version 110C+ User's Manual," May 2005. Air Force SEEK Eagle Office Computational Aeromechanics Team.

2. "Beggar Version 114J Source Code," May 2005. Air Force SEEK Eagle Office Computational Aeromechanics Team.

3. Altmann S. L. *Rotations, Quaternions, and Double Groups*. Oxford: Clarendon Press, 1992.

4. Anderson J. D. *Fundamentals of Aerodynamics* (3rd Edition). New York, NY: McGraw Hill, 2001.

5. Biedron R. T., Vatsa V. N., and Atkins H. L. *Simulation of Unsteady Flows Using an Unstructured Navier-Stokes Solve on Moving and Stationary Grids*. Technical Report 2005-5093, Hampton, VA: AIAA, June 2005.

6. Blazek J. *Computational Fluid Dynamics: Principles and Applications*. Kidlington, Oxford, UK: Elsevier Science LTD, 2001.

7. Brock J. M. and Jolly B. A. *Application of Computational Fluid Dynamics at Eglin Air Force Base*. Technical Report 98-5500, Eglin AFB, FL: AIAA, 1998.

8. Burden R. L. and Faires J. D. *Numerical Analysis* (7th Edition). Pacific Grove, CA: Brooks/Cole, 2001.

9. Cenko A., Phillips K., and Holmes M. *Captive Trajectory System Sting Effects on Store Loads*. Technical Report, St. Louis, MO: McDonnell Douglas Corporation.

10. Coleman L. A., Jolly B. A., Chesser B., and Brock J. M. *Numerical Simulation of a Store Separation Event from an F-15E aircraft*. Technical Report 96-3385, Eglin AFB, FL: AIAA, 1996.

11. Dudley J. G. and Westmorelan W. S. *Mach and Confiugration Effects of a CBU104 WCMD on the F-15E*. Technical Report TEAS Reference No. 401793, Sverdrup Report, January 2004.

12. Garcon F., Taravel P., and Raffin J. C. *Recent Developments in Captive Trajectory Systems of the ONERA Modane Wind Tunnels*. Technical Report 2001-0579, Modane, France: AIAA, January 2004.

13. Goldstein H., Poole C., and Safko J. *Classical Mechanics* (3rd Edition). Reading, MA: Addison Wesley, 1997.

14. Hewson R. *Jane's Air-Launched Weapons* (46th Edition). 2005.

15. Hirsch C. *Numerical Computation of Internal and External Flows*, *2*. New York, NY: John Wiley & Sons, 1984.

16. Hughson M. C. *A 3-D Unstructured CFD Method for Maneuvering Vehicles*. Ph.D. dissertation, Mississippi State University, Mississippi State MS, December 1998.

17. Lee J. M., Dunworth K. S., Rizk M., Westmoreland W. S., and Atkins D. J. *Studies of Combined Use of CFD and Wind Tunnel Test Approaches to Simulate a Store Separation from F-15E Using Efficient CFD Database Generation*. Technical Report 2004-4724, Eglin AFB, FL: AIAA, August 2004.

18. Maple R. C., "AERO 753 Computational Fluid Dynamics Class Notes," 2005. Air Force Institute of Technology.

19. Maple R. C. and Belk D. M. "Automated Set Up of Blocked, Patched, and Embedded Grids in the Beggar Flow Solver." *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields: proceedings of the 4th International Conference held at Swansea, Wales*. April 1994. CD Proceedings.

20. Meyer C. D. *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2000.

21. Murman S. M., Aftosmis M. J., and Berger M. J. *Simulations of 6-DOF Motion with a Cartesian Method*. Technical Report 2003-1246, Moffet Field, CA: AIAA, January 2003.

22. Noack R. W. and Jolly B. A. *Fully Time Accurate CFD Simulations of JDAM Separation From An F-18C Aircraft*. Technical Report 2000-0794, Eglin AFB, FL: AIAA, 2000.

23. O'Neill C. R. and Arena A. S. *Aircraft Flight Dynamics with a Non-Inertial CFD Code*. Technical Report 2005-230, Stillwater, OK: AIAA, January 2005.

24. Power G. D., Calahan J. A., and Hensley D. L. *A Sytem for Moving-Body CFD Simulations on Overset Structure and Unstructured Grids*. Technical Report 2004-716, Arnold AFB, TN: AIAA, January 2004.

25. Prewitt N. C., Belk D. M., and Maple R. C. "Multiple-Body Trajectory Calculations Using the Beggar Code," *Journal of Aircraft*, *36*(5):802–808 (September 1999).

26. Rizk M., Ellison S., and Prewitt N. *Beggar: A Store Separation Predictive Tool*. Technical Report 2002-3190, St. Louis, Missouri: AIAA, June 2002.

27. Rizk M. and Lee J. *Beggar Code Implementation of the (6+)DOF Capability for Stores with Moving Components*. Technical Report 2004-1251, Reno, Nevada: AIAA, January 2004.

28. Samareh J. A. *Application of Quaternions for Mesh Deformation*. Technical Report TM-2002-211646, Langley Research Center, Virginia: NASA, April 2002.

29. Storm K. G. *Validation of Turbulence Models for the Beggar Code in Unsteady Flows*. MS thesis, Graduate School of Engineering, Air Force Institute of Tech-

nology (AETC), Wright-Patterson AFB OH, March 2005. AFIT/GAE/ENY/05-M22.

30. Tannehill J. C., Anderson D. A., and Pletcher R. H. *Computational Fluid Mechanics and Heat Transfer* (2nd Edition). Philadelphia, PA: Taylor & Francis, 1997.

31. Thoms R. D. and Jordan J. K. *Investigations of Multiple Body Trajectory Prediction Using Time Accurate Computational Fluid Dynamics*. Technical Report AIAA 95-1870, AIAA, June 1995.

32. Tuttell R. J. *Enhanced Range Applications Program (EnRAP): Meeting the Future "Truth Source" Needs of the Test Community*. Technical Report 2005-7631, Nashville, TN: AIAA, December 2005.

33. Verhoff A. *Far-field Computational Boundary Conditions for Three-Dimensional External Flow Problems*. Technical Report 96-0892, Saint Louis, MO: AIAA, January 1996.

34. Wertz J. R. *Space Mission Analysis and Design*. El Segundo, California: Microcosm Press, 1999.

35. Westmoreland S. *A Comparison of Inviscid and Viscous Approaches for Store Separations*. Technical Report 2002-1413, Eglin AFB, FL: AIAA, June 2002.

36. White F. M. *Viscous Fluid Flow*. Boston, MA: McGraw Hill, 1991.

37. Whitfield D. L. and Lafe T. K. *Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes*. Technical Report 91-1539, AIAA, June 1991.

38. Zilberman M. and Shay M. *The Use of the Captive Trajectory System for Computation of Trajectories to the Impact Point*. Technical Report 92-4031, Lod, Israel: AIAA, 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)* 23 Mar 06	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* AUG2004 – 23MAR2006

4. TITLE AND SUBTITLE Free Flight Store Simulation Using Beggar	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Babcock, Judson T., 2d Lt, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GAE/ENY/06-M02
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Seek Eagle Office ATTN: Capt Jacob Freeman, 46 SK/SKI 205 W. D Avenue, Ste 348 Eglin AFB FL 32542 DSN: 875-1295	**10. SPONSOR/MONITOR'S ACRONYM(S)** **N/A**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The complete numerical solution of the airflow around a store in extended free flight is of particular importance to the United States Air Force. Beggar is the primary CFD program used by the USAF to obtain solutions for store separations. However, Beggar's ability to simulate a store in free flight is limited because the store must fall through a static background mesh, eventually reaching a point where the solution will fail. The length of any free flight simulation is consequently limited by the height of the background mesh. Code modifications are made to Beggar to remove this requirement by pinning the store in the background mesh at its center of gravity. Rotations are accomplished within the background mesh, but translations are reflected as changes in the grid speeds of the background mesh. This allows the numerical simulation to continue indefinitely. Beggar's ability to model moving components (e.g. control surfaces) in multi-body problems is fully preserved. The modified code is applied to the MK-84 AIR model, which demonstrates that the solution of a pinned store using the modified code adequately matches the solution of a translating store using the unmodified code. In addition, extended free flight simulations are conducted in which the dynamic behavior and long term trajectory of the store are observed. The longest simulation lasts for 135 seconds of solution time. Testing of a generic store body with multiple moving fins results in good agreement between the unmodified and modified solution methods. The modified code reduces overall computational cost by 17% for simulations of similar length because of the smaller background mesh. The combination of indefinite runtime and control surface modeling will make Beggar a powerful tool for studying the non-linear dynamic behavior of stores in free flight.

15. SUBJECT TERMS
Beggar, CFD, MK-84, store separation, mesh motion, background mesh, free flight

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Lt Col Raymond Maple
REPORT **U**	**ABSTRACT** **U**	**c. THIS PAGE** **U**	**UU**	124	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4577; e-mail: raymond.maple@afit.edu